A Method for Concept and Technology Exploration of Aerospace Architectures

A Thesis
Presented to
The Academic Faculty

by

Frédéric Villeneuve

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Aerospace Engineering
Georgia Institute of Technology
August 2007

A Method for Concept and Technology Exploration of Aerospace Architectures

Approved by:

Dr. Dimitri N. Mavris
College of Engineering
Georgia Institute of Technology, Advisor

Dr. Carlee A. Bishop
Georgia Tech Research Institute

Mr. Claude R. Joyner
Pratt & Whitney Rocketdyne

Dr. Daniel P. Schrage
College of Engineering
Georgia Institute of Technology

Mr. Mark H. Waters
College of Engineering
Cal Poly State University

Dr. Alan W. Wilhite
College of Engineering
Georgia Institute of Technology

Date Approved: June 2007

To mom and dad

ACKNOWLEDGEMENTS

This thesis would not have been possible without the support and guidance of my advisor, Dr. Dimitri Mavris. I will always be grateful to him for offering me this great opportunity to study at Georgia Tech. Moreover, I wish to thank the members of my committee, Dr. Carlee Bishop, Mr. Claude Joyner, Dr. Daniel Schrage, Mr. Mark Waters, and Dr. Alan Wilhite. Their feedbacks were indispensable to the realization of this work. Further considerations go to Mr. Mark Waters for his valuable advices and teaching during the course of my graduate studies. He was a great mentor and friend during my years at Georgia Tech. In addition, I would also like to thank Dr. Joseph Saleh for his precious help and feedbacks toward the end of my PhD studies.

My deepest appreciations go to all my colleagues at the Aerospace Systems Design Lab. I feel fortunate to be a member of this great academic organization. Particularly, I wish to thank Patrick Biltgen, Caleb Branscome, Bjorn Cole, Lexie Hautau, Carl Johnson, Michelle Kirby, Jan Osburg, Reza Rezvani, and Cara Zell, for their assistance during the accomplishment of my thesis. I would also like to thank my classmates at the Space Systems Design Lab, Reuben Rorschneider, Zach Trevor, David Young, and Jimmy Young, for their technical assistance in the development of my launch vehicle modeling and simulation environment. I extend my appreciations to Dr. John Olds for his help on space vehicle modeling.

My friends from Québec were also important to the success of my thesis, and I wish to thank them all for their support. My friends from Georgia Tech also receive my dearest gratitude. I wish to thank Simon Briceño, Kyle Collins, Stéphane Dufresne, Ismael Fernández, Mandy Goltsch, Jack Zentner, and Henry Won for their feedbacks,

proofreading, and most of all for those great moments of laughter that made my journey at Georgia Tech so enjoyable.

Last but not least, I cannot say enough about the support of my entire family from Québec. Despite the distance, their love and support considerably helped me in overcoming continuous obstacles. I will ever be grateful to my parents and my brother, who believed in me and always encouraged me to pursue my dreams. My dearest gratitude finally goes to my fiancée, Hai-Mien. Her extraordinary wisdom, patience, and love were instrumental in the accomplishment of this endeavor. I look forward and feel lucky to spend my future in her company.

Frédéric, July 2007

SUMMARY

This dissertation presents the development of a new concept and technology exploration methodology for aerospace architectures. The methodology is based on modeling the design space by a graph, and optimizing the graph using Ant Colony Optimization. The results show that the proposed design methodology can explore more efficiently the concept and technology space of a launch vehicle architecture than traditional optimization approaches such as Genetic Algorithm and Simulated Annealing.

The purpose of the method is to introduce quantitative and simultaneous exploration of concept and technology alternatives during the early phases of conceptual design. To achieve this goal, technical challenges such as expanding the size of the design space, exploring more efficiently the design options, and simultaneously considering technologies and concepts are overcome.

The total number of design alternatives grows factorially with the number of concepts in the design space. Under these circumstances, the design space is difficult to explore in its totality. Considering more alternatives has been the focus of several researchers, using Genetic Algorithms and Simulated Annealing. The large number of incompatibilities between alternatives, however, limits these optimization algorithms and reduces the number of concepts or technologies that can be considered.

To address these problems, a concept and technology selection methodology is developed. The methodology proposes a way to automatically generate aerospace architectures, and to model concept and technology incompatibilities by means of a graph. In conjunction with this new modeling approach, a graph-based stochastic

optimization algorithm is used to efficiently explore the design space. This design methodology is applied to the simultaneous concept and technology exploration of an expendable launch vehicle architecture.

This study demonstrates that the consideration of more design alternatives can help design engineers to make more informed decisions during the concept and technology selection process. Moreover, the simultaneous exploration of concepts and technologies has the potential to identify a different set of solutions than the standard approach where the technologies are explored after the concepts have initially been selected.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS OR ABBREVIATIONS

ACO Ant Colony Optimization.

APAS Aerodynamic Preliminary Analysis System.

AVID Aerospace Vehicle Interactive Design.

BLISS Bi-Level Integrated Systems Synthesis.

CaLV Cargo Launch Vehicle.

CER Cost Estimating Relationship.

CEV Crew Exploration Vehicle.

CO Colloborative Optimization.

CONSIZ Configuration Sizing Program.

DDT&E Design Development Testing and Engineering.

DOE Design of Experiments.

ECD Electrical Conversion and Distribution.

ESAS Exploration Systems Architecture Study.

FASTPASS Flexible Analysis for Synthesis, Trajectory and Performance for Advanced Space Systems.

FLOPS Flight Optimization System.

FPI Fixed Point Iteration.

GA Genetic Algorithm.

GLOW Gross Liftoff Weight.

HAVOC Hypersonic Air Vehicle Optimization Code.

INCOSE International Council on Systems Engineering.

IRMA Interactive Reconfigurable Matrix of Alternatives.

LEO Low Earth Orbit.

LH2 Liquid Hydrogen.

LOX Liquid Oxygen.

LSAM Lunar Surface Access Module.

MDO Multidisciplinary Design and Optimization.

MER Mass Estimating Relationship.

NAFCOM NASA Air Force Cost Model.

NASA National Aeronautics and Space Administration.

OBD Optimization-Based Decomposition.

ODIN Optimal Design Integration.

OEC Overall Evaluation Criterion.

OMS Orbital Maneuver System.

POST Program To Optimize Simulated Trajectories.

RASAC Rapid Access-to-Space Analysis Code.

RBCC Rocket-Based Combined Cycle.

RCS Reaction Control System.

ROSETTA Reduced Order Simulation for Evaluation of Technologies and Transportation Architectures.

RSE Response Surface Equation.

SA Simulated Annealing.

sGA Structured Genetic Algorithm.

SSSP Space Shuttle Synthesis Program.

SSTO Single Stage-to-Orbit.

TFU Theoretical First Unit.

TIES Technology Identification, Evaluation, and Selection.

TPS Thermal Protection System.

TSP Traveling Salesman Problem.

TSTO Two Stage-to-Orbit.

UTE Unified Tradeoff Environment.

CHAPTER I

INTRODUCTION

1.1 Motivation

This thesis is about the development of a new concept and technology down-selection methodology for aerospace architectures. The concept and technology selection of large aerospace systems is a growing concern in aerospace engineering. Engineering firms are required design and build large systems in a short amount of time, and with a large number of possible solutions. Those systems require important monetary investments, which is why the improvement of the selection processes has been the focus of several researchers in the past.

The selection of systems architecture is not a new problem to the engineering world. During the Space Shuttle conceptual design, for instance, design engineers imagined a large number of vehicle architectures. The conceptual design phase lasted almost a decade and a large number of concepts were considered for the design of a unique aerospace system. This is in fact one of the well known examples where the design space had to be broadly explored.

In 2004, the Exploration Systems Architecture Study (ESAS) [1], lead by the National Aeronautics and Space Administration (NASA), explored future space system alternatives. The study evaluated various launch and exploration vehicles to enable the achievement of the future space exploration objectives. The exploration of a large number of systems within a short period required a large engineering workforce. The selected systems will require the investment of several billion dollars, which shows the importance of selecting a good architecture.

The concept selection in engineering fields other than in the space industry is

important as well. Recognizing this issue, the Department of Defense created the Analysis of Alternatives Program to develop new concept selection methodologies. The program's goal is to quickly improve the selection of large military complex systems, and system of systems.

Finally, systems engineering, and the International Council on Systems Engineering (INCOSE), are dedicated to this problem by structuring the architecture selection process. This organization admits that "developing the system architecture is one of the most important responsibilities of the systems engineering." [42] This is because the system architecture decisions have an important impact on the life-cycle of a product and must be done meticulously. It is well documented in the literature that the decisions taken early in the design phase impact the outcome of a design, [47, 56, 85, 105]. Huang, for example, mentions that "80% of the configuration is determined during this phase" [49]. Despite this acknowledgement, the decisions made in the early phases of the conceptual design are difficult to make because of the lack of knowledge.

The examples mentioned above show that systems engineering firms are often faced to a common design problem. This problem is to select the concepts and technologies of complex aerospace systems. This is thus the research focus of my thesis. The next section describes in more detail the problems related to this field.

1.2 Problem Statement

As discussed in the introductory paragraphs, this thesis is centered around the development of a concept and technology selection methodology for complex aerospace architectures. More specifically, the goal is to develop a methodology that enables the quantitative and simultaneous design space exploration of the aerospace architecture concepts and technologies. In other words, the methodology is aimed at adding fidelity and enabling quantitative concept exploration as an aid to design engineers

during the architecture selection process.

This study is important and timely in the context of designing complex systems. First, the number of design options of real-world aerospace vehicles is immense. Considering all the design options and technologies becomes difficult because the number of design alternatives increases factorially with the number of design options. Consequently, the traditional design approach can lead to suboptimal concept and technology selections. Second, this study is timely with the ever increasing system complexities, as well as the limitations in technology funding. Addressing these problems could thus help design engineers to explore the concept and technology space of complex systems.

In light of these problems, this thesis proposes a new design process to address the issues related to the quantitative design space exploration of concept and technology alternatives. This selection methodology enables the quantitative design space exploration of a launch vehicle architecture, simultaneously with a large technology set. This methodology is developed by integrating graph theoretical concepts and Ant Colony Optimization in the conceptual design phase. This approach is also enabled by modeling and providing more data down the top-level components of an aerospace vehicle.

The development of this new method fits into the existing body of knowledge of both the concept selection and technology selection methods. Analyses of past conceptual design projects will show that the common mistake that designers make is to settle on a concept early in the design process [21, 65] even though there is a lack of quantitative data necessary to evaluate the available alternatives. In addition, once a concept is selected, designers tend to pursue this concept rather than generate new alternatives. [32, 67] A typical design philosophy early in the design phase tends to quickly select the architecture and precipitate the design team into engineering analyses.

However, the new method presented in this thesis aims at addressing these concept and technology selection issues by incorporating more quantitative data early in the architecture selection process. This then enables the application of optimization algorithms to search more efficiently over the modeled design space. Design engineers can then use the outputs from the proposed concept and technology selection methodology to make more informed decisions in the selection of the vehicle architecture. The next section shows some background work related to these design problems.

1.3 Background to the Study

Several concept selection methodologies exist for complex aerospace, and mechanical systems. For example, the Pugh matrix [76], the weighted attribute matrix [68], or the morphological matrix [109] have been extensively used in the past. Moreover, the design space exploration of continuous spaces is already in an advanced state with Multidisciplinary Design Optimization (MDO) techniques that explore more thoroughly the design space. [88] The effect of MDO on the design space exploration efficiency is significant, as the different techniques can find more optimal solutions without incurring a large analysis cost. It shows how expanding and optimizing the design space of complex systems can lead to better designs with little or less cost to the overall concept investigation within the architecture study process. Moreover, the introduction of stochastic optimization techniques have enabled the exploration of continuous and discrete design variables [11].

Unfortunately, the exploration of concepts using such techniques is difficult to perform. On the one hand, Pugh or weighted attribute matrices lack quantitative design space exploration capability. They consider a few number of alternatives. On the other hand, MDO lacks design space exploration capability when it is mostly discrete because of the use of gradient-based optimization algorithms, which do not work in presence of discrete, non-numerical design variables. Discrete optimization algorithms

such as Genetic Algorithm (GA), also tend to explore the design space with less efficiency when there is a large number of concept incompatibilities. This problem arises because of the stochastic optimization operations on the design vector does not take into consideration the hierarchy of the system and its concept incompatibilities.[22] The hierarchy is defined here by the structure and the order between the subsystems of a product. For example the design option related to the number of stages of a launch vehicle belongs to a higher hierarchical level than the type of engine within each stage. The use of a penalty function is often necessary, which results in the exploration of sections of the design space that are unfeasible.

For the problems mentioned above, this thesis seeks to enable the quantitative exploration and optimization of the concept and technology space encountered during the early phases of conceptual design. Therefore, we need to improve the process of exploring the large number of alternatives occurring during conceptual design. This need is the starting point for the next section, which describes the purpose of the study.

1.4 Purpose Statement

The purpose of this thesis is to develop a concept and technology down-selection methodology to help design engineers in the down-selection of aerospace vehicle concept and technology alternatives. This goal thus requires an earlier use of quantitative information in the concept and technology down-selection phase, as well as a quicker and more accurate evaluation of the various concepts within the design space.

Purpose Statement

The purpose of this thesis is to develop a design methodology that enables the quantitative and simultaneous exploration of concept and technology alternatives for aerospace architectures

---

The purpose statement is supported by three thesis objectives that are carried out during the course of this dissertation. They are enumerated below:

Objective 1: Expand the size of the design space explored during the concept selection of aerospace architectures.

Objective 2: Improve the means for determining the optimal architecture and to reduce the computation time to reach this optimality.

Objective 3: Improve the simultaneous and quantitative down-selection of concepts and technologies

The first objective addresses the issue encountered during the concept selection phase, where, traditionally, only a small number of design alternatives are considered. As suggested by Finger and Dixon [32], the quality of a design is usually better if a large variety of concepts have initially been considered. Therefore, considering more alternatives should improve the outcome of a design.

The second objective states that, in addition to expanding the design space, the selected alternatives should have a better performance and lower cost than the traditional approach. The design space, however, because of its discrete nature, is difficult to explore. Also, the large number of design incompatibilities encountered during the architecture selection makes the design space exploration even more difficult to perform. This design problem thus justifies the second objective.

Finally, the third objective introduces the notion of simultaneously exploring the technologies and concepts during the architecture selection. This objective is elaborated because of the assumption that technologies and concepts are coupled. Therefore, the selection of certain concepts will lead to a better overall solution if combined with a specific set of technologies. For example, a thermal protection material may be better suited for a capsule concept of an atmospheric reentry vehicle because of the good material resistance to high heat rates, but not well suited for a lifting body

shape because of its poor load-bearing qualities. To avoid this problem, the third objective of this thesis focuses on developing a design method that simultaneously selects the concepts and technologies.

The purpose statement and its three supporting objectives are translated into a top-level research question to guide the remaining of this thesis. It reemphasizes the problem of exploring quantitatively large design spaces populated with a large number of incompatible concepts and technologies.

Main Research Question

What elements are required to improve the optimality of the concept and technology down-selection process during an aerospace architecture study?

In addition to this main research question, three more specific research questions, as well as two hypotheses are elaborated later in Chapter 3. For now, the definition of the problem and the objectives lead to the next section, which describes the significance of this thesis.

1.5 Significance of the Thesis

The concept and technology exploration methodology developed for this thesis is justified by two main elements. The first one is the design process performed during the concept and technology selection of large aerospace architectures. It was mentioned earlier that considering more alternatives leads to better designs. Therefore, a new design method that achieves this goal could benefit the design of real world systems or architectures such as the recently performed space exploration architecture study performed by NASA.

The second intent of the development of this down-selection methodology consists of bringing more insights in the knowledge base of aerospace designs. Similar to the

different techniques in MDO that have improved the state-of-the-art in optimization of multidisciplinary problems, the new exploration methodology proposed in this thesis could be one of the players that contribute to the improvement of the concept and technology exploration within aerospace vehicle architecture studies.

1.6 Delimitations

There is a large number of definitions of a system architecture (or architecture). In this thesis, the term is defined as the arrangement of the elements (physical, operational, maintenance, manufacturing) which define a product through its life cycle in an intent to satisfy its requirements. This thesis is tailored toward the selection of the physical elements for launch vehicle architectures, for which a modeling and simulation environment can quantitatively model the alternatives through the integration of engineering disciplines.

Furthermore, from a design perspective, the formal concept and technology exploration methodology developed in this thesis is applicable to the clean-sheet design phase. In other words, designs for which the system architecture is undefined, and that have a large number of design alternatives and technologies. This happens at the early stages of the conceptual design once the requirements have been defined.

Several examples of the concept and technology selection problems presented in this thesis are concerned with the design of launch vehicles. This emphasis is because launch vehicles are systems that have a large number of concept and technology alternatives, as witnessed during the NASA architecture study [1]. Therefore, the new concept and technology selection process is applied to launch vehicle design. It is, however, applicable to any type of system architecture that can be quantitatively modeled.

Finally, it is important to note that this thesis proposes an alternative approach to down-selecting the concept and technology alternatives during an architecture

study. The methodology thus aims at helping design engineers, which are the integral part of the design process, in making more informed decisions during the aerospace architecture exploration process.

1.7 Organization of the Thesis

This thesis is organized in the following way. Chapter 2 describes the literature review on past conceptual design studies, MDO, concept selection methods, technology selection, and combinatorial optimization. Chapter 3 presents the research methodology and the proposed design space exploration method to alleviate the problems in the selection of complex system architectures. Chapter 4 describes the theory regarding the morphological analysis, graph theory, and Ant Colony Optimization. Chapter 5 describes the implementation of the design process and the different tools enabling it. Chapter 6 presents the modeling and simulation environment developed to demonstrate the effectiveness of the new design method in resolving the design problems during architecture selection. Chapter 7 shows the application of the design method on the simultaneous concept and technology selection problem. Chapter 8 discusses the results, and addresses the research questions and hypotheses proposed in Chapter 3. Finally, Chapter 9 concludes this dissertation with the overall findings, the future works, and the recommendations.

CHAPTER II

LITERATURE REVIEW

This chapter presents a review of the concept and technology selection methods used to explore aerospace architectures. The goal is to look at how design engineers perform concept selection, including the tools they use, and to look at other fields from which theoretical elements could be borrowed to enable a new concept selection method. The chapter is divided in the following seven sections:

1 Conceptual design studies
2 Historical review of architecture studies
3 Multidisciplinary Design and Optimization (MDO)
4 Concept selection methods
5 Technology selection methods
6 Combinatorial optimization
7 Benchmarking

2.1 Design Studies

2.1.1 Conceptual Design Studies

The problem of selecting an aerospace architecture is not new to the field of aerospace engineering. This section presents some concept selection examples performed during the early design of launch vehicles.

Chase [14] studied various vehicle shapes, number of stages, propulsion types (air-breathing and rocket), advanced materials, and trajectory types (vertical vs horizontal takeoff and landing). The figures of merit are the vehicle weight, cost per pound, and operations cost. Each architecture is designed independently and its performance is compared according to quantitative design criteria such as performance, risk, cost, and technological uncertainty. Some interesting observations can be made from the

study. For example, the study shows that reusable vehicles are more cost-effective than the Space Shuttle. It also shows that a launch vehicle with airbreathing propulsion has better performance but is more costly than rocket vehicles because of its development cost.

Similar to Chase, Freeman *et* al. [34] explores single stage, two stage, rocket propulsion, airbreathing propulsion, wing body, and conical launch vehicle body concepts. In addition, near-term and advanced technologies are among the alternatives analyzed for this study. The disciplinary models used to assess the performance are: Solid Modeling Aerospace Research Tool (SMART), the Aerodynamic Preliminary Analysis System (APAS), the Program to Optimize Simulated Trajectories (POST), and the Configuration Sizing Program (CONSIZ). This multi-disciplinary environment enables a proper quantification of the vehicle performance, which is important in selecting an aerospace architecture. The study shows the importance of considering technologies while selecting launch vehicle architectures.

Dorrington [30] performs a qualitative study to select the future launch vehicle architecture for the European Space Agency. Various launch vehicle alternatives are considered such as rocket and airbreathing propulsion, the number of stages, reusability, vertical or horizontal takeoff, and also included a pool of available technologies. The conclusions are that launch vehicle architectures are difficult to compare qualitatively, yet the various launch vehicle options must be analyzed concurrently using self-consistent methods to select the proper launch vehicle architecture.

As way to explore a larger number of concepts, Rockwell International Science Center [10, 78] created *Design Sheet* to enable the quick design space exploration of aerospace systems. This computational tool applies a set of optimization and constraint management algorithms that explore physics-based and historically-regressed parametric equations modeling aerospace system, and it enables a more global design space exploration. The design engineer can thus perform quick sensitivity analyses

in the early phases of conceptual design within a short setup time. For example, Bowcutt *et* al. [7] use *Design Sheet* to model airbreathing architectures. Their goal is to determine the technological and propulsive needs for affordable access-to-space vehicles. The authors show nine airbreathing vehicle architectures and compare them to seven other rocket-based vehicle architectures. This example shows that a significant number of alternatives can be analyzed with a quick modeling and simulation tool.

2.1.2 Technology Studies

On the technical end, Caluori *et* al. [13] performed a technology study on a single and two stage launch vehicle. The authors evaluated various launch vehicle technologies from composite structures, to dual fuel engines, extended life engines, vehicle configuration changes, and slush propellant. In addition the authors show that "*technology findings are sensible to the vehicle concept*", where the technologies had a more important impact on the single stage vehicle. The study's main outcome is that accelerating technology advancements can significantly improve the vehicle performance and cost. The report, however, do not discuss technological uncertainty.

Haefeli, *et* al.[40] performed a technology assessment on a single stage reusable launch vehicle. He followed a four step approach which consists of: 1) determining the past technology growth from historical data, 2) performing the preliminary design of various launch vehicles (essentially three single-stage launch vehicles), 3) forecasting technology uncertainty using the historical data and the expert analysis, and 4) determining the technologies for which funding should be allocated to improve the vehicle performance. The study enables the determination of critical and high-yield technological areas that should be funded. The results of the study shows a ranking of different technology programs performance according to figures of merits related to the vehicle weight, its life cycle cost, and the Research and Development (R&D)

cost. The authors also showed that several propulsion programs had a potential for net loss. Moreover, reusable surface insulation materials, propellant tanks, gimbal nozzles, and integration engineering were found as critical technology programs.

Similarly, Hepler and Bangsund performed a technology study by forecasting the technology performance in the future (seven years) and evaluating their impact on four single-stage launch vehicle configurations.[44] The outcome of the study recommended further development of technology areas, as well as the reassessment of vehicle performance.

2.1.3 Sizing and Synthesis Programs

Evaluating the performance and cost of such design studies required the development of modeling and simulation environments. Different modeling philosophies are available and presented in the following paragraphs. These models were used as a starting point for the development of the sizing and synthesis tool, described in Chapter 6.

The Optimal Design Integration (ODIN) system was developed in the 1970's as a sizing and synthesis programs for reusable launch vehicles. It is composed of a library of independent disciplinary programs (aerodynamics, thermodynamics, propulsion, weights, structure, aeroelasticity, cost, stability), controlled by an executive program (ODINEX) that exchange data through a database [35]. The program enables the quick setup of launch vehicle design problems and the automated design space exploration using reliable disciplinary modules. However, ODIN is difficult to setup when dealing with innovative concepts because of geometry modeling problems. [104]

Similar to ODIN, the Aerospace Vehicle Interactive Design (AVID) is composed of a library of disciplinary programs, an executive program, and a database allowing data management between the disciplinary programs. This approach enables easier integration of the disciplinary models by sharing common design parameter through a database. Among the disciplines modeled in AVID are the geometry, aerodynamics,

propulsion, weight, performance, and economics. They are integrated into an interactive design environment which facilitates the interaction with the designer during conceptual and preliminary design. A large diversity of vehicle concepts can be evaluated with the program, including winged, multi-stage, horizontal landing, and vertical landing launch vehicles[104].

The Space Shuttle Synthesis Program (SSSP) [70] was developed for assessing the implications of design changes on the Space Shuttle during its conceptual design. It has the capability of modeling various two-stage reusable launch vehicle concepts, and to perform sensitivity studies on trajectory configurations and design variables. SSSP consists two general disciplinary programs, General Trajectory Simulation Module (GTSM) and weight/volume program, and it emphasizes modeling simplicity and minimum input requirement, thus limiting design flexibility compared to generalized sizing and synthesis programs.

The Hypersonic Air Vehicle Optimization Code (HAVOC) is a FORTRAN written code. It models single and multi-stage, winged, airbreathing, and rocket launch vehicles. It uses a parametric geometry representation based on arbitrary body parameters such as enhanced super ellipses [69], which are then input to the structural analysis, aerodynamics, and weight disciplinary models. It also has its own trajectory simulation model, and uses a reduced version of the NASA Air Force Cost Model (NAFCOM) for modeling economics.

More recently, the Flexible Analysis for Synthesis, Trajectory and Performance for Advanced Space Systems (FASTPASS)[96] was developed as a need for improved optimization of the vehicle configuration and mission requirements. Its goal is to automate the optimization process occurring during the vehicle sizing and performance evaluation. The program uses, such as AVID and ODIN, a synthesis database to share the data between disciplines. Its disciplinary models (aerodynamics, propulsion, trajectory, structures, and weights) are developed for the program specifically, yet it

can accept other models programmed in the FASTPASS language, which permits to adapt the discipline fidelity to the modeling requirements.

The Reduced Order Simulation for Evaluation of Technologies and Transportation Architectures (ROSETTA) is a sizing and synthesis environment for launch vehicles based upon user-generated metamodels[19]. It is spreadsheet-based and it requires a specific metamodel for each architecture modeled. The metamodels are generated from higher fidelity tools such as APAS, POST, and CONSIZ [25, 75]. Its fast computation time enables the exploration of thousands of vehicle configurations to consider design parameters uncertainty.

In summary, the launch vehicle design tools presented above show that highly coupled, high-fidelity disciplinary modules are required to perform an accurate performance assessment of launch vehicle concepts. The use of MDO and metamodels is also benefiting the design engineers by exploring more efficiently and more quickly the design space. This approach will be used in the creation of the modeling and simulation environment, detailed in Chapter 5.

2.2 NASA Vehicle Design Programs

This section reviews the problems and challenges observed during the conceptual design of two programs: the Space Shuttle and the NASA Exploration Systems Architecture Study (ESAS). This exercise is aimed at observing the requirements for the development of a concept and technology selection approach.

2.2.0.1 Space Shuttle

The Space Shuttle is a well-documented space program. Its flight history, the description of its systems, its flight performance as well as its entire design process are publicly available. This section thus describes an overview of the Space Shuttle design process and emphasizes on determining the challenges that occur during the conceptual design of large aerospace architectures.

The Space Shuttle design process took place over a period of 20 years, from 1960 to 1980. A thorough description of the conceptual design process, and the evolution of the Space Shuttle architectures was carried out; it is described in Appendix A. The following paragraphs present what are considered the most important observations to be made from the Space Shuttle experience.

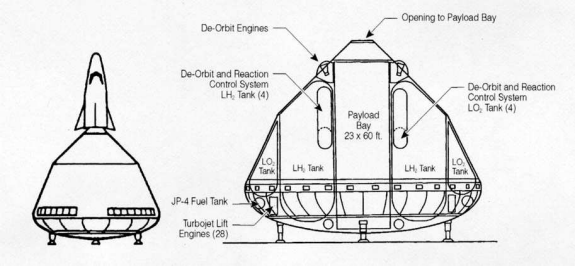
To start this discussion, several design problems were observed during the Space Shuttle conceptual design process. The first problem observed was that the program requirements changed several times over the conceptual design phase. For example, the payload weight, dimensions, and type, as well as the mission requirements were constantly modified over time. At the beginning of the conceptual design phase, the shuttle had to deliver military and scientific satellites into orbit, fly reconnaissance missions, and transport sections of the International Space Station for its construction. By the end of the 60's, the military payloads and reconnaissance missions were abandoned because of the withdrawal of the U.S. Air Force from the program. In addition to the mission requirements, the budget was constantly reduced. This fluctuation in funding caused considerable design changes to reduce the vehicle's development and production costs. This constant change in requirements observed during the design process justifies the need of a quick concept and technology down-selection methodology, which can be iteratively performed in line with requirement changes. In other words, different requirements can change the solution in the design space, which requires that a new concept and technology down-selection processes must be performed under different requirements.

Next to the evolving mission requirements, another design challenge observed during the Space Shuttle design process was the difficulties incurred by a long design phase, lasting over a decade. It was difficult to keep the core design team in place and to keep track of the decision rationales. As a result, the concept selection process seemed unorganized. Considering, the size of the design task, this lack of organization

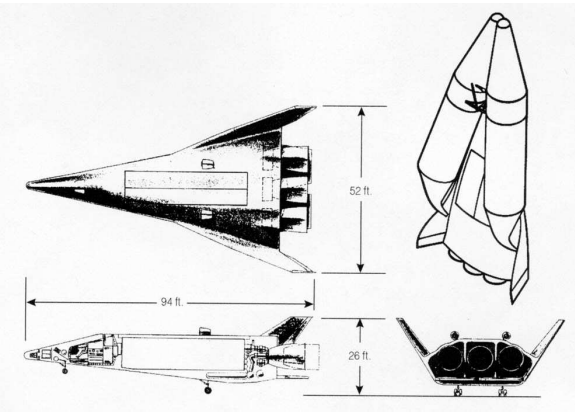
resulted in a disordered Space Shuttle design evolution, where some concepts dropped earlier in the design phase were later brought back as design alternatives. For instance, NASA reinstated the stage-and-a-half concept toward the end of the conceptual design phase, yet it had been dropped earlier [45]. This justifies the need of a concept selection methodology that can simultaneously explore all the quantifiable concepts in the design space, thus avoiding the premature elimination of potential concepts.

Besides the decision rationales, the evaluation of the different alternatives was also difficult to perform. Various concepts were independently explored without cross-fertilization between them. Cross-fertilization is referred to the exchange of concepts between architectures. Each contractor pursued one configuration at a time, carrying out their analyses with their own set of design tools. For example, George Muller, a NASA officer, criticized the lack of breadth in the design space exploration: "various groups and designers were still pursuing their individual approaches [45]." Furthermore, the orbiter and booster were designed separately. In fact, NASA designed the orbiter first because of its technical challenges and then designed the booster. This task could have been greatly improved, had the selection of both stages been done simultaneously.

Another problem encountered during the Space Shuttle conceptual design was the concept evaluation approach. In short, NASA awarded several design contracts to the main space companies, which used different modeling and simulation environment [45]. The design reviews thus compared Space Shuttle concepts using different modeling assumptions, comparing apples with oranges. This thesis proposes to perform the down-selection of aerospace vehicle architectures using a single modeling and simulation environment to avoid this problem. This modeling environment, described in Chapter 6, is developed as a means to demonstrate the concept and technology down-selection methodology using publicly available data, as well as the benefit of using a single sizing and synthesis program for concept and technology down-selection.



(a) Chrysler booster



(b) Lockheed Starclipper

Figure 2.1: Space Shuttle early concepts. Source: Heppenheimer [45]

The design freedom is another characteristics observed during the Space Shuttle design process. This allowed NASA's engineers to a large variety of concepts. From single stage rocket to airbreathing propulsion, via reusable and expendable concepts, the organization thoroughly explored the design space. This design philosophy probably helped to generate concepts as creative as the Chrysler booster, shown in Figure 2.1a, and the Star-Clipper, shown in Figure 2.1b. The need of preserving the design freedom during the design process thus justifies the need of a flexible modeling and simulation environment that can quantitatively model a large variety of concepts and that allows the easy addition and removal of concepts.

Overall, this review of the Space Shuttle design process is useful to determine some characteristics needed in the development of the concept selection methodology presented in this thesis. These are 1) preservation of design freedom by enabling the easy addition and removal of concepts and technologies in the process, 2) concept

comparisons with the same modeling tool, and 3) rapidity of execution of the down-selection process. As shown in the previous paragraphs, a concept and technology down-selection methodology with these characteristics could have helped the Space Shuttle design engineers in providing additional information to support their concept and technology down-selection process.

2.2.0.2 NASA Exploration Systems Architecture Study (ESAS)

The Space Shuttle fleet will retire after 2010, which creates the need for another manned launch vehicle. The conceptual design and concept selection of this vehicle was performed during the Exploration Systems Architecture Study (ESAS).

This design project, initiated in 2004, called for the development of the future space system architecture to enable the U.S. space exploration vision to return to the Moon and send humans to Mars. And, similar to the Space Shuttle, the ESAS has left a well-documented evolution of its conceptual vehicle designs. The review of this source of information helps us understand how different is the conceptual design process today and what are the major problems still remaining.

A large number of vehicle designs were performed during the ESAS, such as the lunar lander, the cargo delivery vehicle, the Crew Exploration Vehicle (CEV), and the crew and cargo launch vehicles. As observed from the ESAS final report, the concept of architecture design was exploited during the study. Since the ESAS involved a large number of systems, a thorough architecture study was performed. For example, Figure 2.2 presents a reference mission architecture for lunar exploration.

The large number of systems involved in the exploration architecture complicates the design space exploration of the space mission architecture. For launch vehicles, NASA's approach consisted of selecting the concepts based on qualitative and quantitative figures of merits. First, a trade tree was drawn to determine the available

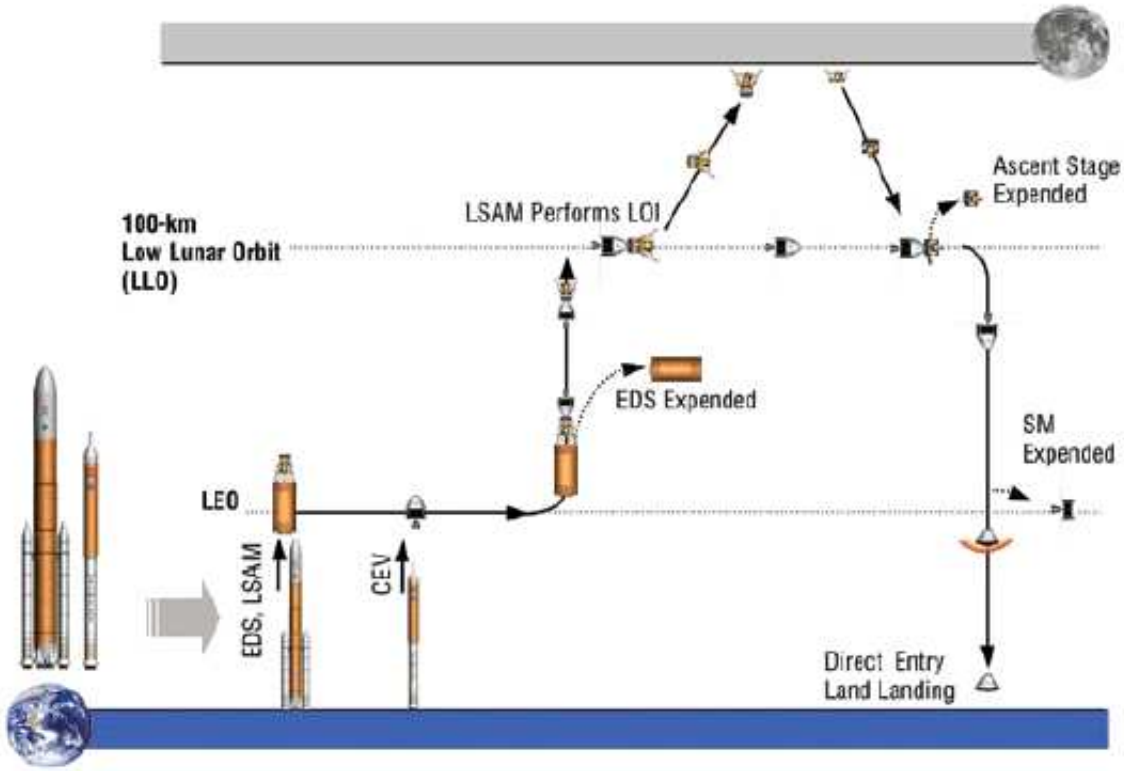


Figure 2.2: ESAS reference mission architecture. Source: NASA [1]

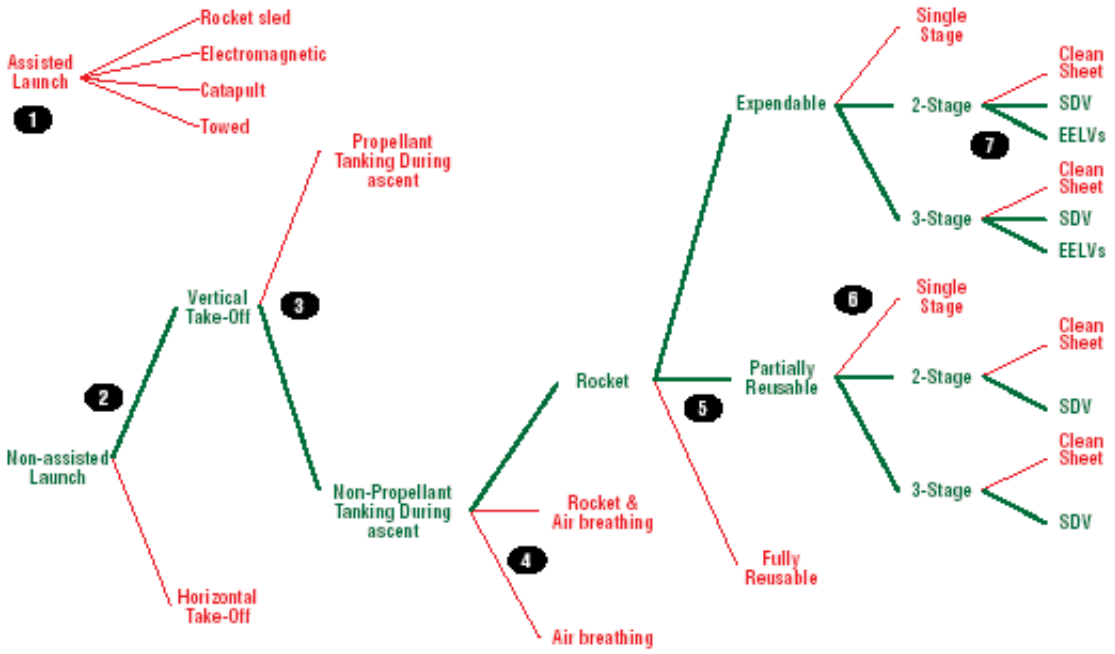
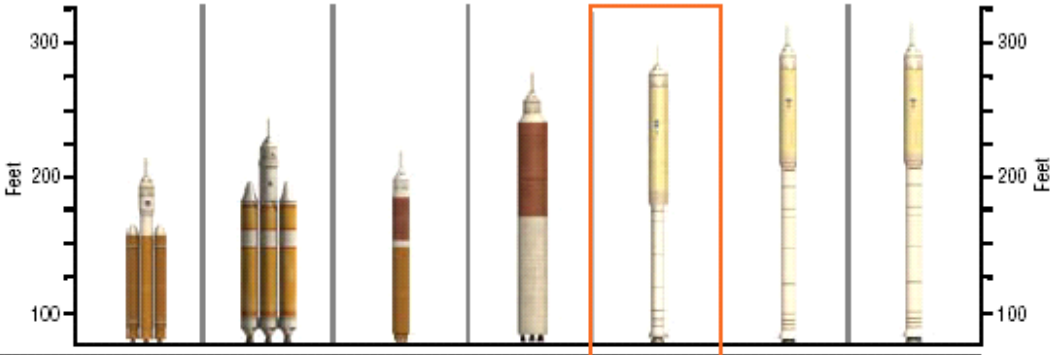


Figure 2.3: ESAS launch vehicle trade tree concept selection. Source: NASA [1]

options. Figure 2.3 shows an example of launch vehicle trade tree. Once these possibilities were drawn, NASA down-selected the concepts based on the technology readiness, low development cost, high reliability, and budget availability. For this reason, the general architecture selected consisted of a vertical takeoff, multistage, expendable launch vehicle with rocket (solid or liquid) propulsion systems.

Once the general launch vehicle architecture selected, the ESAS team went into more detailed concept exploration where different types or stages, propellant, and rocket engines were evaluated. For example, among the strap-on booster options considered were the Atlas-V RD-180, the Delta-IV RS-68, and the Zenit; in addition, rocket engines such as the SSME, the J-2S, the RS-68, or new expended cycles were considered. Every design option was then evaluated, sized, and compared. The most promising concepts for the CLV and the CaLV are displayed in Figures 2.4 and 2.5, respectively. The figures show the various concepts explored for the launch vehicles, their respective level of performance, and the ultimately selected concept.



	Human-Rated Atlas V/New US	Human-Rated Delta IV/New US	Atlas Phase 2 (5.4-m Core)	Atlas Phase X (8-m Core)	4 Segment RSRB with 1 SSME	5 Segment RSRB with 1 J-2S	5 Segment RSRB with 4 LR-85
Payload (28.5°)	30 mT	28 mT	26 mT	70 mT	25 mT	26 mT	27 mT
Payload (51.6°)	27 mT	23 mT	25 mT	67 mT	23 mT	24 mT	25 mT
DDT&E*	1.18**	1.03	1.73**	2.36	1.00	1.3	1.39
Facilities Cost	.92	.92	.92	.92	1.00	1.00	1.00
Average Cost/Flight*	1.00	1.11	1.32	1.71	1.00	.96	.96
LOM (mean)	1 in 149	1 in 172	1 in 134	1 in 79	1 in 460	1 in 433	1 in 182
LOC (mean)	1 in 957	1 in 1,100	1 in 939	1 in 614	1 in 2,021	1 in 1,918	1 in 1,429

LOM: Loss of Mission LOC: Loss of Crew US: Upper Stage RSRB: Reusable Solid Rocket Booster

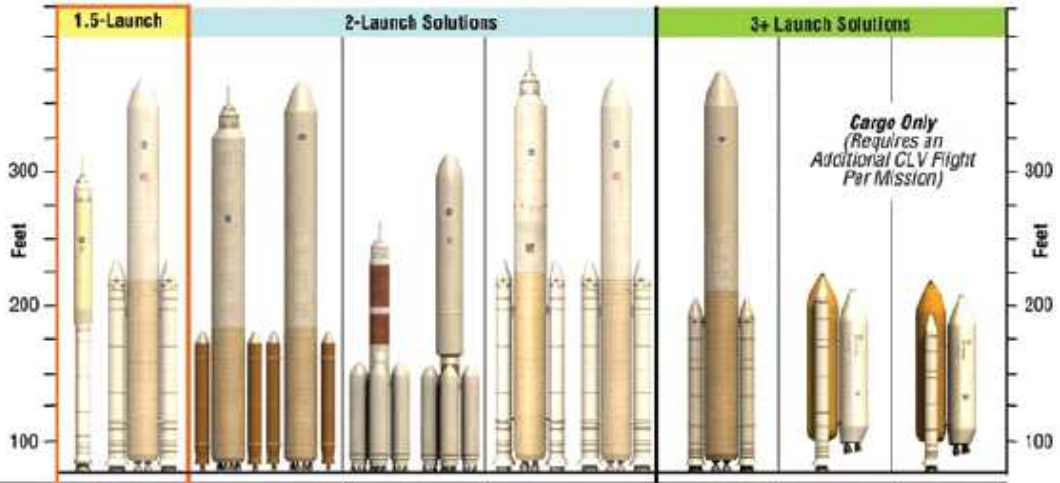
Figure 2.4: Crew launch vehicle architecture comparison. Source: NASA [1]

The Exploration Systems Architecture Study showed that the concept and technology down-selection process is often performed under a tight schedule (concept and technology down-selection performed in less than two years) and requires the quantitative evaluation of a large number of vehicle concepts. Therefore, a concept and technology down-selection methodology emphasizing rapidity, and enabling the exploration of a large number of concepts could help design engineers in performing concept tradeoffs more quickly.

2.3 Multidisciplinary Design

2.3.1 Traditional Design Processes

Conceptual design of aerospace vehicles has received a lot of attention since the remarkable growth in powerful computational modeling and simulation capabilities. It is a phase of the design process that is highly multidisciplinary in nature because of



	5 Segment RSRB In-Line with 5 SSME Core - Cargo	Atlas Phase X (8-m Core)	Atlas Phase 3A (5.4-m Core)	5 Segment RSRB In-Line with 4 SSME Core	4 Segment RSRB In-Line with 3 SSME Core	5 Segment RSRB Side-mount with 3 SSME	4 Segment RSRB Side-mount with 3 SSME
Payload to 28.5	106 mT (125 mT w/ upperstage)	95 mT	94 mT	97 mT	74 mT	80 mT	67 mT
Lunar LV DDT&E*	1.00	1.29	.59	.96	.73	.80	.75
CLV+Lunar Crew/Cargo DDT&E (family op)*	1.00 (4 Seg RSRB w/1 SSME)	1.26 (Atlas V)	1.02 (Phase 2)	.98 (4 Seg RSRB w/1 SSME)	.83 (4 Seg RSRB w/1 SSME)	1.03 (5 Seg RSRB w/J2S-LR85)	.85 (4 Seg RSRB w/1 SSME)
Lunar LV Facility DDTE	1.00	1.33	2.25	1.00	* All cost estimates include reserves (20% for DDT&E, 10% for Operations), Government oversight/full cost; Average cost/flight based on 6 launches per year. **Production costs are higher than in-line due to production of separate side-mount cargo carrier		
CLV+Lunar Facilities Cost	1.00	1.12	1.56	1.00			
Lunar LV Average Cost/Flight*	1.00	1.08	1.19	.87	.78	1.13**	1.13**
LOM – Cargo (mean)	1 in 124	1 in 71	1 in 88	1 in 133	1 in 176	1 in 172	1 in 173
LOC (mean)	1 in 2,021	1 in 536	1 in 612	1 in 915	1 in 1,170	N/A	N/A

Figure 2.5: Cargo launch vehicle architecture comparison. Source: NASA [1]

the involvement of several engineering disciplines. Figure 2.6 shows the work-flow of the traditional design process as observed by Rowell *et* al. [85]. It is composed of three basic elements: ellipses representing the design decisions, rectangles representing the engineering analyses (also called disciplinary analyses), and arrows representing the flow of information between the engineering analyses and design decisions. The engineering analyses serve to model the vehicle performance and cost, whereas the design decisions represent the different alternatives available to a decision maker during the

conceptual design process.

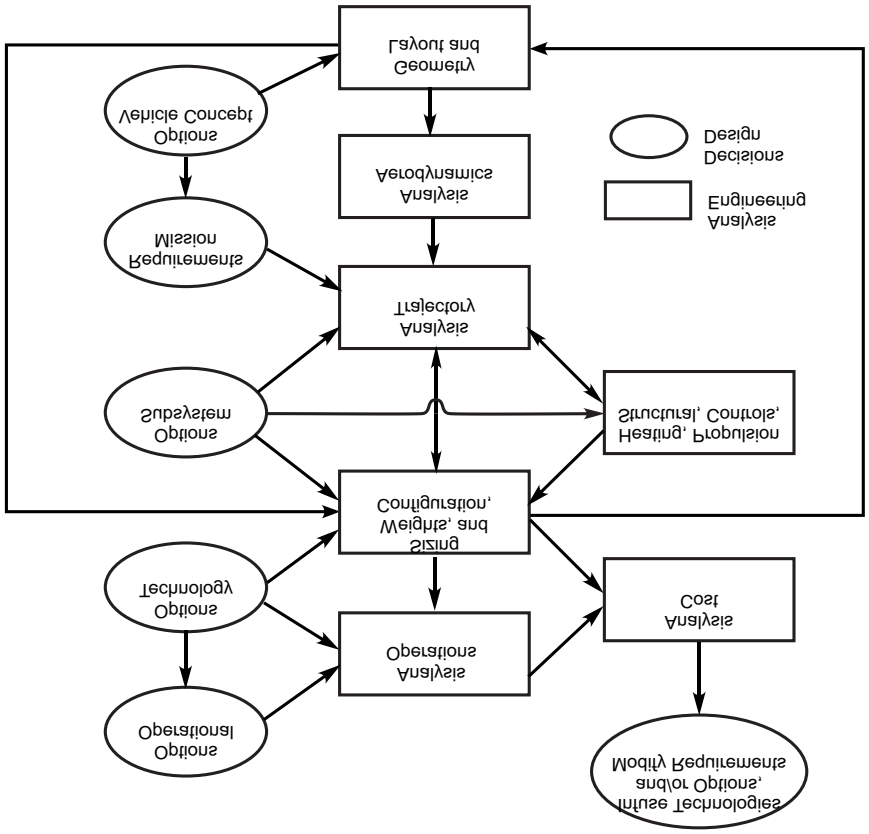


Figure 2.6: Traditional conceptual design process. Adapted from Rowell *et* al.[85]

An interesting observation, that can be made from the design process overview in Figure 2.6, is that the vehicle operational and disciplinary concept alternatives, mission requirements, and technologies directly impact the engineering analyses. The decisions taken at this level influence considerably the design decisions. Moreover, it is observed that the disciplinary analyses are highly coupled, and the design process is highly iterative. These characteristics impact considerably the decision-making process because they increase both the computation time, as well as the time required to set up a multidisciplinary environment between each engineering analysis. The development of a concept selection process must thus take these constraints into considerations as they influence the number of alternatives that can be evaluated.

Unlike MDO, the architecture selection process has received less attention despite

its importance in the outcome of the design task probably because of its difficulty to be performed within a structured framework. For example it is only recently that architecture selection processes are proposed by organization such as the INCOSE. The difficulty of selecting an architecture arises from the nature of the design space, which evolves from a discrete nature during the architecture selection phase to a continuous design space later in the conceptual design. Discrete design spaces are highly combinatorial and therefore difficult to explore. The total number of combinations increases factorially with the number of design options which makes their exploration almost impractical. As an example, a vehicle composed of 15 subsystems, having five (5) concepts per subsystem, has 5^{15} possible combinations of architectures. Attempting to explore this combinatorial design space with the traditional launch vehicle design process is therefore impossible.

2.3.2 Optimization

Until now, the focus of the discussion was on the description of the design process performed in the aerospace industry. The next sections, however, discuss the various design tools and methods developed to address these issues and improve the design process occurring in conceptual design. The first set of methods discussed is optimization.

Multidisciplinary Design Optimization (MDO) is a design methodology that has considerably improved the conceptual design process despite its relatively recent emergence [85]. It is defined by Sobieski as a *"methodology for the design of systems where the interaction between several disciplines must be considered, and where the designer is free to significantly affect the system performance in more than one discipline [88]."* This design philosophy improves the conceptual design process by accelerating the convergence rate, and finding more optimal solutions in the design than traditional man-in-the-loop design approaches. For example, Bowcutt [6] obtains a 46% increase

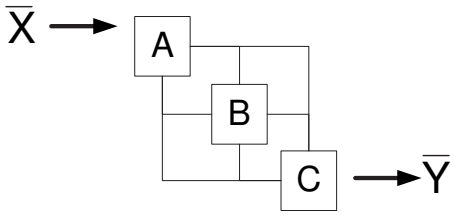


Figure 2.7: Notional design structure matrix of a fixed point iteration for three disciplinary analyses

in the range of a hypersonic missile by applying MDO to the propulsion, aerodynamics, stability, trajectory, and mass properties models, as opposed to the manual optimization approach.

The standard approach in optimizing the disciplines of complex aerospace vehicles consists of linking them together in a Fixed Point Iteration (FPI) approach, as shown in Figure 2.7. This figure illustrates a notional example of solving multidisciplinary problems, where A, B, C represent the disciplinary analyses, and X and Y the initial design vector and the final design vector, respectively. This problem, although simplistic, shows that the various disciplines are coupled by sharing common design variables, which must be optimized. Reorganization of the disciplinary analyses, can sometimes reduce the number of feedback loops but it is usually difficult to eliminate them completely.

Optimizing an aerospace vehicle with the FPI approach can be time-consuming and difficult to converge because of the feedback loops that require inner sizing loops. Some other techniques were developed to solve this problem. Could these techniques help during the selection of concepts and technologies? The next paragraphs provide an answer to this question.

One of the ways to facilitate the optimization of a vehicle made of highly coupled disciplines such as shown in Figure 2.7 is using Optimizer Based Decomposition (OBD). This MDO approach handles globally the local constraints of each engineering discipline involved in the performance estimation of the vehicle. It thus reduces the

number of iterations by exploring globally the design space and creates a smoother design space. Braun et al. [8] shows that the optimization time can be three to four times faster with OBD over FPI for the design of a SSTO launch vehicle with 36 design variables. The vehicle is indeed optimized three to four times faster with OBD.[1] However, this approach can hardly be applied to concept selection since it has difficulty dealing with large discrete design spaces.

Similar to OBD, Collaborative Optimization (CO) is a decomposition technique that parallelizes the integration of disciplinary analysis codes. It accelerates the overall optimization process by eliminating the inner loops as well as the coupling between disciplines. The disciplinary modules thus manage their own design constraints and they perform their own internal optimization. The design variables are optimized by a top-level optimization algorithm, which also controls the simulations performed by each disciplinary module. As stated by Braun *et* al. [8], a better handling of the coupling between disciplines has a significant impact on the efficiency of exploring the design space. The authors show the benefits of CO on a SSTO dual-fuel rocket launch vehicle by reducing the computation-time by a factor of three over a standard FPI technique. Unfortunately, such as OBD, CO is difficult to apply during concept selection because of the impossibility of using gradients on discrete design options.

Finally, the Bi-Level Integrated System Synthesis (BLISS) is a two-step approach that separates the subsystem, and the system variables. The first step consists of optimizing each vehicle discipline individually by freezing the system-level variables. For example, by fixing the vehicle thrust-to-weight ratio, the vehicle optimizes its geometry, and engine parameters, and so on. The second step consists of optimizing the system-level variables using a gradient-based method. For this step, the thrust-to-weight ratio is optimized by letting the vehicle engine and geometry parameters

[1]The authors refer to OBD as sequential compatibility constraint and to FPI as iterative-loop approach

remain constant. This cycle is repeated until convergence is attained [89]. BLISS has showed to improve significantly the optimization process of subsonic business jets [90], structural optimization on a hub structure [91], and aeroelastic wing design [23]. Although BLISS is not directly applicable to the concept exploration of aerospace architectures, the method shows that decoupling an independent optimization of the subsystems is possible when the disciplinary analyses are synthesized and the vehicle optimized periodically during the design process. This finding is essential to the justification of the concept selection method formulated in the next chapter.

In summary, the MDO techniques are usually performed to optimize vehicles modeled by computationally-expensive disciplinary models. The integration of those disciplines in a MDO problem thus requires the use of efficient optimization algorithms such as the gradient-based optimization models. The MDO approach thus assumes that the design space is mostly continuous and without local minima. For this reason, the MDO techniques mentioned in this section can hardly be used in the concept exploration phase. The next section surveys established concept selection methods.

2.4 Concept Selection Methods

As previously discussed, MDO techniques are focused mainly on the optimization of design spaces with continuous design variables. There is, however, a set of techniques that handles more easily discrete design spaces formed by the concepts and technology options. These techniques are presented in this section.

2.4.1 Morphological Matrix

One of the concept selection approaches aimed at handling the discreteness of design options is the morphological matrix. The morphological matrix is a systematic means of representing the design options of a complex system. It is derived from the morphological analysis, which explores the relationships between the various functions of a complex problem [81]. Zwicky first developed the morphological analysis as a

means of structuring the relationships between the various elements of astronomical problems [108]. The technique is summarized in this section as being directly used in the design methodology developed in Chapter 3.

Although the first use of this method was in astrophysics, this approach is applicable to conceptual design since it enables the organization of a vehicle's solutions in groups with similar characteristics. For example, Table 2.1 presents the morphological matrix for a notional aircraft architecture. One can observe the overall structure of the morphological matrix where the first column represents the morphological field (or functions) of the problem and each row in the second column represents a set of concepts that can perform the function. A system is completed by selecting a concept from each row of the morphological matrix. In addition, from the observation of Table 2.1, no assumptions are made on the type of vehicle. It can be a rotorcraft, an aircraft, or an airship, for example. This characteristic of the morphological matrix is the reason why it improves the creativity and enhances the chances of finding innovative solutions [108]. The goal of this section is to review some of the interesting applications of the morphological matrix, and a more detailed description of the approach is presented in Chapter 4.

Table 2.1: Morphological matrix for a notional aircraft problem

Morphological field	**Concepts**		
Takeoff Approach	Accelerated rolling	Sling shot	Dropped
Sustain its motion	Fixed Wing	Rotorary wing	Buoyancy
Landing Approach	Rolling	Diving	Crashing

The morphological matrix can represent the alternatives of complex aerospace systems. For example, Hollingsworth and Mavris [48] use the matrix to organize both the technologies and the concepts of hypersonic cruise missiles. A total of 33 subsystems are used for regrouping all the different vehicle concepts. The authors manually select the alternatives from the morphological matrix without quantitatively exploring them.

The quantitative design space exploration of the morphological matrix using modeling and simulation tools is not an easy task. In fact, it is only recently that Buonanno [11] has shown that the morphological matrix, coupled to a structured Genetic Algorithm, can be explored using a modeling and simulation environment. The author explores alternatives of a supersonic transport aircraft with only including compatible alternatives. Buonanno also shows that a stochastic optimization approach can be coupled to a morphological matrix to facilitate its exploration. The problem, however only included 576 compatible combinations of alternatives, which is well under the number of combinations found in a typical system.

The exploration of a morphological matrix populated by incompatible alternatives is indeed not an easy task. As a response to this problem, Weber and Condoor [103] used the Theory of Coupling and a hierarchical function definition procedure. This procedure is as follows:

1. Identify a set of independent primary functions
2. Identify the alternatives for each function
3. Select the alternatives
4. Repeat (1) with the secondary functions

This hierarchical approach thus uses the "hierarchical nature of the design process" [103] and repeats the above approach until the concept is satisfactory. This approach thus ensures the generation of compatible alternatives because the functions are always independent of each other. However, the approach cannot be automated in a general design process.

Dealing with incompatibilities is necessary when performing the exploration of concept and technology alternatives. Indeed, some of these alternatives are incompatible, therefore cannot be selected at the same time. To attack the problem of dealing with alternative incompatibilities, Kirby and Mavris [57, 58] propose the use

of the "Technology Compatibility Matrix" for technology incompatibilities. It consists of developing a pairwise compatibility between the elements of the matrix where each row or column represents a technology. The approach, however, does not model the hierarchy found between concepts.

The automatic generation of concepts from the morphological matrix is a challenge addressed by many researchers. For example, Ritchey [81] used a computerized morphological analysis combined with a pairwise consistency matrix to analyze threat scenarios for the Swedish Defense System. His approach required a human to select the concepts, but the morphological and incompatibility matrices are coded in a program to let the design engineer explore the design space. Similarly, an Interactive Reconfigurable Morphological Matrix (IRMA) [64] was developed at the Aerospace Systems Design Lab of Georgia Tech. The matrix, considers incompatibilities between concepts, and enables the filtering of concepts with respect to their technical readiness and their system-level performance. For example, alternatives may be selected with respect to their endurance, weight, volume, or power required. Finally, Strawbridge *et* al. [107] developed a concept generator based on the morphological matrix. This approach shows that a computational means of generating concepts can explore larger design spaces. In other words, the computer usually generates a larger number of concepts than humans usually do because humans have limited capacity to analyze complex combinatorial spaces.

In summary, the morphological matrix is a powerful concept generation approach that facilitates the manipulation of large combinatorial spaces. However, the morphological matrix only helps in generating concepts, not in selecting them. The next sections presents the down-selection approaches for large number of alternatives.

2.4.2 Decision Matrices

The decision matrix is a second type of concept selection approach, after the morphological matrix. Unlike the morphological matrix that helps to generate and organize alternatives, decision matrices help to down-select from an initial pool of alternatives. Two popular forms are commonly found in engineering design: the Pugh matrix and the weighted attribute decision matrix.

The Pugh matrix consists of comparing the concepts to a baseline from a predetermined set of attributes [76] (e.g., takeoff gross weight, overall cost, noise, technological risk). Each concept is rated as better, same, or worse than the baseline for each attribute. The best concepts will be those with the highest number of better ratings. A notional example of the Pugh matrix is presented in Table 2.2, where the signs + and − represent a concept that is better or worse than the baseline, respectively. The best concept according to this method is the concept C. Despite its simplicity, potential concepts can be eliminated from the Pugh matrix because of the scoring technique, which is described in the next paragraph.

Table 2.2: Example of a Pugh Matrix

		Concepts		
	Baseline	A	B	C
Attribute 1	Same	+	+	+
Attribute 2	Same	-	+	-
Attribute 3	Same	-	-	+
Attribute 4	Same	-	-	+

The weighted attribute matrix is similar to the Pugh matrix, except that each attribute is weighed for its importance. Each concept is rated numerically and the alternative that gets the highest weighted sum is considered the best alternative. The Pugh matrix presented in Table 2.2 is transformed into a weighted attribute matrix in Table 2.3 with the same four attributes and three concepts. The total score at the bottom is the sum of the individual scores for each concept multiplied by the weight

of each attribute. According to this method, the concept B is better than A and C.

Table 2.3: Example of a Weighted Attribute Decision Matrix

		Concepts		
	Weight	A	B	C
Attribute 1	3	4	5	5
Attribute 2	4	2	4	1
Attribute 3	5	2	2	3
Attribute 4	1	1	2	4
Total		31	43	38

Although simple to use, these concept selection methods have two major drawbacks: 1) they can eliminate potential concepts when they are desirable, and 2) the final results are highly dependant on the weights and ranking assigned by designers, introducing bias in the selection. The first drawback is due to the Pareto frontier problem, which will be discussed in Section 2.4.4, but to solve this issue, Mullur *et* al. [68] used *Compromise Programming* and a preference structure in a weighted attribute matrix. This approach minimizes the designer's personal perception and avoids the elimination of desirable concepts.

2.4.3 Stochastic Optimization Algorithms in Concept Selection

This third category of concept selection methods regroups the stochastic optimization algorithms, which refer to methods that involve randomness in the optimization process. The two most popular stochastic optimization algorithms used for concept selection are Genetic Algorithms (GA) and Simulated Annealing (SA).

Genetic Algorithms have served well a large number of aerospace vehicle design problems, where discrete and continuous design variables are usually mixed in the design space exploration. For example Perez *et* al. [74] explored the design space of an aircraft, using three discrete and 13 continuous variables with a GA. Similarly, Blasi *et* al. [5] explored the design space of an aircraft, using nine design variables with a GA, where only one was a discrete variable. Crossley *et* al., [21] use GA to

optimize a 50-seat aircraft, where the engine type (turbofan and turboprop), number of engines, and engine locations where the possible alternatives. The authors used the Flight Optimization System (FLOPS) as the sizing and synthesis code and a penalty function to model design constraints.

The concept selection is different than a traditional design space exploration, yet stochastic optimization algorithms seem to explore as well as the continuous design space. For example, Mosher [67] uses a GA to find the optimum combination among 20 concepts and 432 combinations for the design of a spacecraft vehicle. The concepts are related to the energy management type, structure, propulsion and launch vehicle. Jilla and Miller [53] use SA to explore design options of distributed satellite systems.

Despite its wide range of applicability mentioned in the paragraph above, GA often has poor performance for the concept and technology selection. This is because of the incapability of modeling the concept hierarchy in the genetic population, [22] and the incapability of modeling concept incompatibilities.

The incapability of modeling the concept hierarchy arises because GAs randomly construct their solutions. There is no logic in the solution construction. Hence, Dasgupta and McGregor [22] modified the algorithm to create a structured Genetic Algorithm (sGA). Their approach enables the use of binary redundancy and hierarchy in the bit string. In other words, the sGA creates a chromosome hierarchy, where lower level genes of the chromosome will be active only if their higher level genes are active. This method was applied successfully on the exploration of 576 supersonic aircraft alternatives, in addition to several other continuous variables [11]. This example thus shows the importance of considering the system hierarchy in the concept selection.

The incapability of considering incompatibilities between alternatives requires that GAs use penalty functions, which makes the algorithm convergence more difficult. This is because the genetic operations (mutation, cross-over, and reproduction) do not always generate compatible solutions. The difficulty in dealing with incompatible

alternatives increases proportionally with the number of incompatibilities in the design space. Typical design problems have morphological matrices with a large number of concept incompatibilities, making thus their use with a GA impractical. The same logic can be applied to SA where the population changes occur randomly, which does not enable the exploration of a large concept space with incompatibilities.

Nevertheless, stochastic optimization algorithms usually perform well in conceptual design. They can optimize continuous and discontinuous design variables, and they can explore more globally the design space because of their capacity of dealing with multi-modal design spaces. Finally, they greatly facilitate the design space exploration of multi-objective design problems by retaining near-optimal solutions in the population base [15]. The topic of multi-objective optimization is discussed in the next section.

2.4.4 Multi-Objective Optimization in Concept Selection

Multi-objective optimization is another research thrust that receives considerable attention. One of the keystone elements in multi-objective optimization is the concept of Pareto optimality. A Pareto optimal solution is a hyper-plane in a design space where the simultaneous improvement in all objectives cannot be achieved. Along with this concept of Pareto optimality, Mattson and Messac [63] developed a Pareto-based concept selection approach, named thereafter s-Pareto. This concept enables the visualization of the Pareto frontier for various design alternatives by generating a Pareto front for each design alternative. Figure 2.8 shows an example of s-Pareto frontier with three alternatives, represented by their respective design space, and two objective functions. Mattson and Messac applied their idea to the selection of a truss structure composed of four different concepts. In addition, McManus *et* al. [65], applied the s-Pareto concept selection approach to satellite constellation selection, Villeneuve [101] to the architecture selection of launch vehicles, and Buonanno [12]

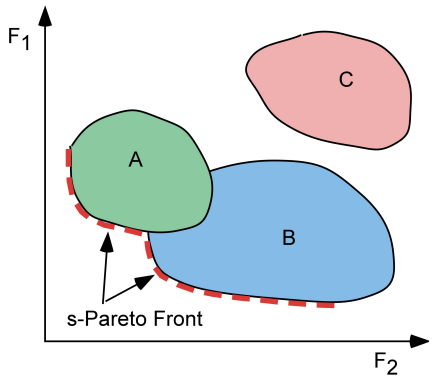


Figure 2.8: Notional s-Pareto frontier example

and Chung [16] to the selection of supersonic aircraft alternatives.

One problem found in multi-objective optimization using GAs is the loss of genetic diversity, which results in premature algorithm convergence. To address this problem, Smaling and de Weck [87] developed a fuzzy Pareto, which means that a band solutions are retained in the Pareto frontier rather than a single Pareto line. This approach preserves genetic diversity. It was applied to the selection of six hydrogen-enhanced internal combustion engines. The approach demonstrates the improvements in preserving design richness and assessing concepts robustness.

In summary, multi-objective optimization algorithms are well suited for the concept selection of a small number of alternatives. s-Pareto optimal solutions, however, are more difficult to determine when the number of objectives increases. Moreover, the computational power required to generate the Pareto fronts increases proportionally with the number of alternatives, which limits the s-Pareto approach to evaluating only a small number of alternatives. Finally, s-Pareto plots do not encourage cross-fertilization between concepts. A set of architectures is selected *a priori* and does not interact with the concepts found in another alternatives.

2.5 Technology Selection Methods

As discussed earlier, two challenges occurring during the early phases of conceptual design consist of 1) selecting the concepts, and 2) selecting a portfolio of potential technologies that will enhance the performance of the actual system under design. This section thus reviews some of the technology exploration and selection approaches developed to date.

The development Technology Identification, Evaluation and Selection (TIES) methodology, by Kirby et al. [57, 58] was a response to the growing need of selecting technologies with a limited funding in research and development. The approach consists in creating Response Surface Equations (RSE's) of the technology impacts on the vehicle performance and cost, and then selecting the set of technologies that maximize an objective specified by the designer. The method uses compatibility matrices to select the appropriate technology portfolio. For example, Kirby and Mavris applied the methodology to the selection of 11 technologies for commercial and high speed civil transport aircraft [56, 57]. The authors used a multi-attribute selection technique, and then selected the best combination of technologies from a full-factorial design of experiments. One of the strengths of the TIES methodology resides in its well-structured and defined design process.

One problem that the TIES methodology has is its difficulty in finding optimal solutions in the presence of a large number of technologies. This problem arises because of the number of technology combinations increases factorially with the number of technologies. It thus becomes more difficult to use, or even create, a computationally manageable Design of Experiments (DOE) size. Unlike DOEs, stochastic optimization algorithms such as genetic algorithms and simulated annealing usually perform well in the exploration of combinatorial design spaces. For this reason, TIES was upgraded with a GA technology exploration algorithm, which enables the exploration over a more efficient combinatorial design space. [83] This improvement increased the

total number of technologies included in the analysis.

The major problem with stochastic optimization algorithms is their difficulty in handling incompatibilities between design variables and technologies. In the case of technology selection, this requires that technology incompatibilities be modeled with penalty functions [84], which affects the convergence rate of the optimizer. In response to this problem, Raczynski *et* al. [77] developed a gene-corrected genetic algorithm for the selection of a technology portfolio among 36 possible technologies. The method corrects the population chromosomes by means of internal logic, programmed in the GA code. The approach shows significant improvement in the number of function calls and a better final objective function compared to traditional penalty-function GA's. This shows how the efficiency in searching (exploring) the design space improves the convergence rate of genetic algorithms.

Using a GA as well, Utturwar *et* al. [100] developed a bi-level inverse design technology selection approach. First, the optimum value of the technology impact (also called k-factors) is found by optimizing a RSE with a gradient-based optimization algorithm. Second, a GA explores the discrete design space formed by the technology combinations. This second step is done to find the technology set that will have the closest setting of the design parameters to the settings previously determined in the first step. Despite not showing any improvement in the objective function, the bi-level approach exhibits significant time-saving and more flexibility during the technology selection process. Nonetheless, this approach cannot explore well design spaces with several minima because of the gradient-based optimization scheme. It thus precludes the choice of inverse design for the concept and technology exploration of aerospace architectures.

The technology selection is sometimes closely related to the vehicle configuration optimization as well as to the requirements definition. In addition, it is important for design engineers to visualize the various technology tradeoffs against the design

objectives. For this reason, Baker and Mavris [2, 3] created a decision-making environment that enables the visualization of the impact of the requirements, vehicle characteristics, and technologies on any complex engineering system. The approach, named the Unified Tradeoff Environment (UTE), resulted in the visualization environment of five mission parameters, five vehicle attributes, and 19 technologies for a Future Transport Rotorcraft. The UTE thus enabled the visualization of the technology impacts according to the design objectives. This usually helps the designer to gain more knowledge about the technology space.

Stanley [92] also proposed a technology engineering framework, which concurrently integrates technology and the system developments. The process manages technology development through five steps: 1) develop of the technology requirements, 2) prioritize and pre-select technologies, 3) plan the technology program, 4) monitor technology programs through figures of merit and milestones, and 5) integrate technologies into the system. The framework was applied to development of a new space transportation architecture composed of a crew vehicle and reusable launch vehicle. It enables better management of technology development, as well as better integration between technology and system developments.

Thus far, the technology selection has significantly improved because of the integration of stochastic optimization algorithms. However, certain problems arise with their use. First, as mentioned in the previous paragraph, stochastic optimization algorithms have trouble dealing with incompatibilities between the design variables. This problem is usually bypassed by tweaking the construction of solutions, or by using a penalty function, but it does not allow an efficient exploration method. Second, there exists no methodology that performs the technology and concept selection simultaneously. This problem is important because technologies and concepts are highly coupled. Thus, the exploration of various concept-technology combinations can potentially improve the design, and specifically when the number of technologies

and concepts increases. One of the fields that has considerably helped the dealing with large combinatorial spaces is a field of mathematics named combinatorial optimization. It is thus the subject of the next section.

2.6 Combinatorial Optimization

The literature reviewed thus far has focused mostly on concept and technology selection techniques in aerospace engineering. Combinatorial optimization is a field that studies problems that are similar to the technology and concept selection. Those problems are discussed in this section, as well as some of the combinatorial algorithms developed to optimize them.

Several combinatorial optimization algorithms exist to find the solution of large discrete problem spaces [46]. These algorithms are effective over a wide variety of problems [24, 99]. In addition, they are applicable to several problem classes. The goal of this section is to present some problems and optimization algorithms that are similar to those we find in aerospace engineering design.

2.6.1 Combinatorial Problems

Combinatorial problems usually have the characteristics that they are simple to describe, yet complicated to solve [79]. There is a large number of categories of combinatorial problems in mathematics representing various aspects of combinatorial spaces as shown below. Specifically, three types of problems show some similarities with the concept and technology selection problem presented earlier: 1) Traveling Salesman Problem (TSP), 2) the job assignment problem, and 3) the network flow problem. These problems are similar to the concept selection problem as they require the selection of discrete elements, and they are detailed in the following paragraphs.

The first problem discussed is one of the oldest combinatorial problems that exists, that is, the TSP. This problem consists of finding the shortest path for a salesman to visit x number cities, and return to the initial city where he started. Figure 2.9

shows, for instance, the optimal solutions for a TSP of 532 U.S. cities. This problem is modeled by an undirected graph where there is a geographic location for each city.

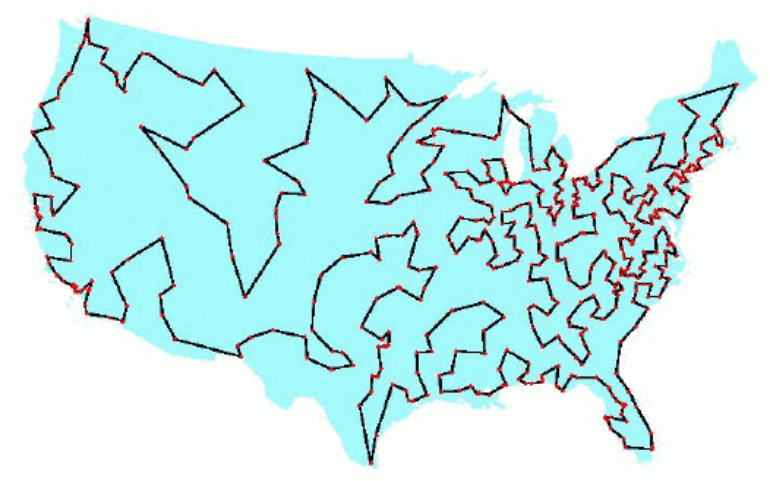


Figure 2.9: Solution of the Traveling Salesman Problem (TSP) for 532 U.S. cities. Source: www.tsp.gatech.edu [18]

The problem's complexity has evolved with the advent of the computer. In 1930, the mathematicians could solve the TSP for 30 cities, whereas in 2004 the problem was solved for 24,978 cities.

The assignment problem consists of assigning X workers to Y tasks, with a cost associated for each worker-task pair. The goal is to minimize the overall cost by assigning a worker on every task. This problem is usually modeled by a bipartite graph, which is a graph where its nodes can be divided in two groups, as shown in Figure 2.10.

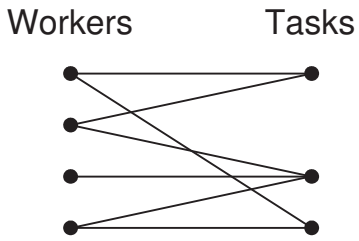


Figure 2.10: Bipartite graph example

Finally, the third combinatorial problem of interest is the network flow problem.

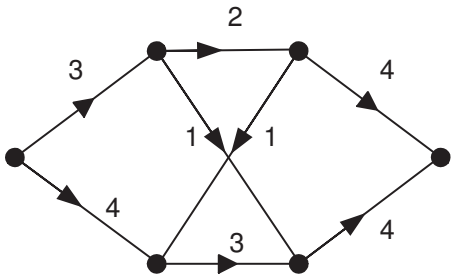


Figure 2.11: Directed graph example with varying edge capacities

This problem consists of finding the path that maximizes the flow over a network with varying capacities along each edge. This problem is useful in traffic management, electronic circuit design, and pipeline optimization. Similar to the two other combinatorial problems presented above, network flow problems are modeled by graphs. But unlike them, they are modeled by directed graphs, where the move from one node to the other can only be done in one direction and has a maximum flow capacity. Figure 2.11 shows an example of directed graph modeling a notional network.

Although there exist many more combinatorial problems in mathematics, two major observations are worth making from this brief overview. The first observation is that the three problems discussed above are similar to the selection of concepts or technologies, where both fields deal with large combinatorial spaces. The second observation is that combinatorial optimization problems are modeled by a graph. This approach seems to facilitate the understanding and the visualization of the problem and to produce better solution methodologies. Because of these similarities, it is believed that the modeling of the morphological matrix with a graph will enable the quantitative design space exploration of the morphological matrix in a same manner as these notional combinatorial problems. The next section discusses different approaches used to solve these combinatorial problems.

2.6.2 Combinatorial Optimization Algorithms

Efforts in combinatorial optimization have also lead to the emergence of several techniques for optimizing these combinatorial problems. Some of these approaches are listed below.

Similar to the engineering field, the combinatorial optimization field has seen good performance using genetic algorithms but less performance using simulated annealing [17]. Since SA and GAs are widely used and understood for engineering application, the interest of this section is to look at some other combinatorial optimization techniques. More specifically, the goal is to look at combinatorial optimization techniques that work well in the optimization of graphs since it was determined in the previous section that a graph could easily model a morphological matrix.

Branch and Bound [61] is a graph-optimization technique that solves smaller regions of the problem, and then bounds those regions together to provide a solution of the bigger problem. For example, for the TSP, the method determines subgroups of good city paths and then assembles these groups to solve the entire problem. In addition to the TSP, the Branch and Bound method works well for assignment problems, as discussed by Lawler and Wood [61]. The approach, however, is not suited to solving problems with incompatibilities and hierarchy because of the lack of logic during the bounding process.

Another graph-based optimization approach is the Ant Colony Optimization (ACO). This optimization algorithm imitates the behavior of ants in a colony when they search food sources. The algorithm has generally a better performance than SA or GA for the optimization of graph-based problems such as the TSP, and the job scheduling problem [17]. In addition to its good performance, ACO can construct compatible solutions for each path. This advantage is because ACO keeps in memory the past moves and construct a solution by following a graph, which model the problem incompatibilities. This characteristic thus justifies its choice over GA or SA, which do not

generate always compatible solutions. It is also different from the branch and bound approach where the solutions created for the subregions of the problem may not be all compatible when they are bounded together. For these reasons, ACO seems to be the algorithm of choice for technology and concept selection.

2.6.3 Combinatorial Optimization and Aerospace Engineering

In aerospace vehicle design, combinatorial optimization has barely been exploited. Patel *et* al. [73] used a network graph theory and physical programming to model the discrete design space of five (5) system alternatives for each of the fuselage, wing, and engine systems of a passenger aircraft. The authors use the k^{th} shortest path algorithm to explore the feasible solutions. The different solutions are selected based on a greedy algorithm.[2] In more details, the fuselage section is selected by maximizing the payload weight, the wing is selected by maximizing the lift-to-drag ratio, and the engine is selected by maximizing the cruise thrust. However, as mentioned in Colorni *et* al. [17], greedy algorithms can lead to poor solutions and hinder the optimization algorithm efficiency. In addition, this modeling approach of the problem is not applicable to the modeling of the morphological matrix because no information is available between the combination of two concepts. In other words, the solution must be constructed in totality before computing its performance. ACO does not require this heuristic information and it is detailed in Chapter 4.

The performance of ACO is sensitive to the value of its input parameters, such as GA or SA. To alleviate this problem, Sun and Teng [94] applied an integer-coded GA and an ant colony optimization algorithm for the layout optimization of the distribution of objects in a communication satellite. The authors show that the two methods are perfectly applicable for this specific problem, yet they suggest that a local search algorithm could improve the algorithm convergence.

[2] A greedy algorithm is referred to as a method of selecting local decision rules in the hope of obtaining a global optimum

With aerospace engineering design problems becoming more complex, combinatorial optimization algorithms could help in the exploration of the design space. This thesis is an example of such a vision.

2.7 Benchmarking

2.7.1 Design Process Evaluation Criteria

Similar to aircraft design, the development of a design process requires the elicitation of some evaluation criteria, which is used to evaluate the process alternatives. Four evaluation criteria are determined, as shown in Table 2.4.

Table 2.4: Design process evaluation criteria

Criteria	Description
1	Execution time
2	Convergence error
3	Capability to explore quantitatively a matrix of alternatives with incompatibilities
4	Adaptability to new concepts and technologies

The first criterion refers to the computation time for the algorithm to converge to a final value. The second one refers to the difference between the optimized value of the algorithm and the real answer (or the best-known answer). Some problems do not have a known solution so in that case, the convergence error is the difference between the optimizer output and the best know solution. The third criterion refers to the capability to explore quantitatively a matrix of alternatives, where some of the alternatives are incompatible. An alternative in this thesis is defined by any concepts or technologies that can be selected. Finally, the fourth criterion refers to the adaptability of the process to the introduction of new concepts or technologies in the design exploration loop. These criteria are used in the next section to compare some of the concept and technology selection approaches reviewed in the previous sections.

2.7.2 Benchmarking

The goal of the literature review is to identify the strengths and weaknesses of the available aerospace architecture exploration approaches, as well as to identify the gaps where new improvements can be made for this phase of the conceptual design. This section thus presents the benchmarking of the relevant concept and technology selection approaches against the evaluation criteria defined in the previous section for design processes. The goal of this benchmarking approach is to quickly identify the gaps in the literature. Five methods are selected and are evaluated on a scale from A to E, where A is a good and E is a bad. The selected methods and their respective performance according to the evaluation criteria (see Table 2.4) are shown in Table 2.5.

Table 2.5: Benchmarking of the concept selection methods

Method	**Author(s), year**	**Criteria**			
		1	2	3	4
Interactive sGA	Buonanno, 2005 [11]	B	A	C	C
Subsystem selection	Mattson & Messac, 2002 [63]	A	C	C	A
IRMA	ASDL, 2006 [64]	E	E	D	B
Decision Matrix	Pugh, 1996 [76]	A	E	D	A
TIES\GA	Raczynski & Kirby, 2003 [77]	A	C	C	C

The first design methodology benchmarked is an interactive structured genetic algorithm applied on the design space exploration of a supersonic business jet. The method is developed by Buonanno [11], and it is described in Section 2.4. This method performs the best under these criteria since it enables a quantitative design space exploration of the matrix of alternatives. It uses a structured GA which has good exploration performance in a discrete design space (Criteria 1 and 2) and can be adaptable to the use of new concepts by adding discrete design variables. However, the method cannot explore alternative incompatibilities, which is why it receives a poor score on the fourth criterion.

The second method is a design selection approach using a network and combined to local measures of merit to select the subsystem options for a notional transport aircraft. [73] It has a fast execution time (Criterion 1) but a poor convergence error (Criterion 2) because it uses the k^{th} shortest path, a greedy algorithm. Greedy algorithms usually exhibits poor performance in large combinatorial spaces. In addition to its poor performance for large combinatorial space, the approach cannot explore incompatible alternatives, explaining the poor score for exploring incompatible alternatives (Criterion 3). Finally, because it uses graph theory, new concepts can be added and explored successively to the design space by updating the graph, which explains its good adaptability to new alternatives (Criterion 4).

The third method is the Interactive Reconfigurable Matrix of Alternatives (IRMA) discussed in Section 2.4.1 [64]. It is an Excel-based morphological matrix that allows the selection of concepts with consideration of their incompatibilities and technology readiness levels. The method has an almost instantaneous execution time (Criterion 1) considering its low complexity in programming. However, it cannot explore quantitatively the concept performance (Criterion 3), yet it considers incompatibilities. Finally, the IRMA is easily adaptable to new concepts (Criterion 4), but it cannot be evaluated on its convergence error (Criterion 2) because the method does not compute the concepts quantitatively.

The Decision Matrices, explained in Section 2.4.2, are simple ways to compare alternatives. For the same reason as the IRMA, Decision Matrices can be executed rapidly (Criterion 1), yet they do not optimize the design space quantitatively (Criterion 3), and cannot be evaluated on the convergence error (Criterion 2) since quantitative analysis is not possible. In regards of their simplicity, they are more adaptable to new concepts (Criterion 4) than the other combinatorial optimization approaches, hence their good scoring performance for the fourth criterion.

The Technology Identification, Evaluation and Selection (TIES) methodology, enhanced with a genetic algorithm (see Section 2.5) is the fifth benchmarked methodology. As described earlier, it enables the technology portfolio selection through the creation of Response Surface Equations (RSEs). The methodology has a fast execution time (Criterion 1) yet a poor convergence error because the algorithm optimizes RSEs. In addition, TIES can evaluate the concept quantitatively but not with incompatibilities (Criterion 3), and it difficult to integrate new concepts or technologies (Criterion 4) because the RSEs usually must be recreated.

Benchmarking the methods relevant to the concept and technology selection problems closes the literature review (Chapter 2). The next chapter (3) describes the research approach and design methodology developed to address the problems mentioned in Chapter 2.

CHAPTER III

RESEARCH APPROACH & METHODOLOGY

Chapter 2 showed that several design techniques have been developed to improve the conceptual design process, yet few were developed to improve the architecture selection process. Chapter 3 now proposes a concept and technology exploration methodology in response to the architecture selection problems of aerospace vehicles.

3.1 Overview of the Traditional Concept Selection Approach

The traditional concept selection approach consists of individually exploring a certain number of architectures, qualitatively selected, and then picking the best from this initial pool of architectures. An illustration of the traditional practice is represented by the approach depicted in Figure 3.1. For this figure, the first step consists of selecting potential architectures, based on the design engineer's judgment and experience. The second step then consists of individually exploring the design space of these architectures. This task is performed by finding the best settings of the design variables for each architecture. Finally, the third step is the selection of the best architecture based on the comparison of their Pareto fronts.

This chapter proposes a different concept (and technology) selection approach. The next section presents the research questions and hypotheses that helped the development of this approach.

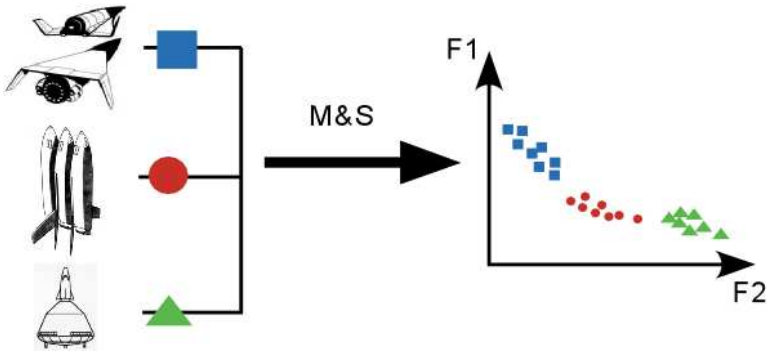


Figure 3.1: Traditional process for the exploration of aerospace architectures

3.2 Research Questions and Hypotheses

The challenges related to the concept and technology exploration of aerospace architectures are not simple to overcome. To address this problem, a set of research questions and hypotheses are therefore developed to help the organization of the present research, and to determine the specific areas that this thesis will focus on.

As mentioned, the selection of aerospace architectures is usually characterized by a design space with a large number of concepts (or alternatives). For example, the morphological matrix of a typical aerospace vehicle usually has hundreds of concepts that represent the subsystems. However, incompatibilities, hierarchy between concepts, and the total number of combinations increasing factorially with the number of design options make this design space difficult to explore during the early phase of conceptual design. The first research question addresses this challenge.

Research Question 1: How can the design space of an aerospace vehicle be more broadly explored during its concept selection?

One of the major difficulties for a quantitative concept exploration of aerospace architectures resides in the discrete nature of the design space, formed by the concepts and technologies. The traditional design approach used during the architecture selection consists of modeling each concept or technology decision in a discrete but

cartesian space. A value of 0 is thus attributed to an inactive concept or technology, and a value of 1 is attributed to an active one. This approach is simple to implement in any discrete optimization algorithms. However, there usually exist a hierarchy and incompatibilities between concepts in early conceptual design. For example, if one is looking at solid or liquid rocket engines, then some concepts such as the engine cycle (e.g., expander, staged combustion, gas generator) cannot be selected when the stochastic optimizer selects a stage that is powered by a solid rocket. This nature of the design space makes its exploration difficult when a design engineer wants to quantitatively explore the concepts. There must be some kind of hierarchical concept selection procedure that must be followed by the discrete optimization algorithm.

This problem is similar to the traveling salesman problem presented in Section 2.6, which is easier to explore when it is modeled by a graph. It is thus hypothesized by the first hypothesis that a graph can model the hierarchy and incompatibilities in a morphological matrix, and help the construction of compatible solutions. The similarity between the two problems should help the construction of physically compatible solutions by modeling the concept and technology alternatives using a graph, which is the topic of the second hypothesis. This ensures a design space exploration in the feasible space and avoids checking the feasibility of the architecture after their creation. To evaluate this hypothesis, it is proposed to compare the feasibility of the solutions generated by the graph approach and the standard random approach. This comparison should provide supporting evidences to prove the first hypothesis.

Hypothesis 1: Modeling the morphological matrix by a graph accelerates and facilitates the construction of physically compatible solutions.

A better solution construction, however, does not guarantee the attainment of an optimal solution. Despite being modeled by a graph, the optimization of any combinatorial and discrete design space is not a trivial task. As mentioned before,

the number of combinations in the design space grows factorially with the number of design options and alternatives. For a complex aerospace vehicle, the number of combinations can reach the billions, which makes the design space computationally intractable. The second research question addresses this problem by investigating whether it is possible to explore this combinatorial space within a respectable amount of time using standard computational resources?

Research Question 2: What design exploration approach could reduce the exploration time to reach the optimality of the solution when exploring aerospace system architectures?

There exist several types of discrete optimization algorithms. For the concept and technology selection problem, the algorithm must be able to optimize a graph, which constitutes the first requirement in an attempt to answer this second research question. This eliminates the possibility of using Genetic Algorithms, Simulated Annealing, or Particle Swarm Optimization because they do not typically deal with graphs. Then, of the available graph-based optimization algorithms, the algorithm selected must be able to perform a hierarchical solution construction. This requirement thus eliminates the Branch and Bound and dynamic programming algorithms[61] because they perform the optimization by finding local solutions in the design space and bounding these local regions together to solve the bigger problem. This, however, does not guarantee the construction of compatible solutions. Moreover, other graph optimization algorithms such as physical programming [73] do not work on exploring the morphological matrix because they require heuristic information to move over the graph. For the Traveling Salesman Problem [18], the distance between two cities is the heuristic information used to select the next move.

It was mentioned in Section 2.6.1, however, that the hierarchical solution construction approach used in Ant Colony Optimization (ACO) has the capability to deal with

concept and technology incompatibilities, as well as the design hierarchy that exists in real-world aerospace architectures. This is done by keeping in memory the algorithm moves, which results in respecting the hierarchy and compatibilities within a given concept. ACO is the only discrete optimization algorithm found by the author to enable this solution construction approach. Therefore, it is hypothesized that Ant Colony Optimization should find more optimal solutions than traditional stochastic optimization algorithms. A detailed description of the algorithm is presented in the next chapter.

Hypothesis 2: Ant Colony Optimization can find more optimal solutions than traditional stochastic optimization algorithms in exploring a morphological matrix.

The technology selection is also coupled to the architecture selection. This implies that the system is more optimal if the concepts and technologies are simultaneously explored. However, simultaneously exploring a pool of technologies and concepts increases the design space size factorially, which makes it harder on the design space optimization algorithm to find optimal solutions. This issue is addressed by the following research question.

Research Question 3: What optimization approach enables a better simultaneous concept and technology exploration of aerospace architectures ?

To answer this question, it is assumed that there is a way to efficiently explore the large number of concepts found at the beginning the conceptual design process. However, Research Question 3 addresses the issue that technologies and concepts are separately selected during conceptual design, yet they both are coupled. Simultaneously Exploring the concepts and technologies increases the size of the combinatorial space. This third research question thus addresses how to solve this challenge.

3.3 Proposed Solution and Methods

3.3.1 New Design Method

Thus far, the focus of the discussion has revolved around the problems and challenges occurring during the process of concept and technology selection in conceptual design. This section represents a turning point of the dissertation, where the focus of the discussion goes from identifying the problems to proposing a solution. For this reason, the new design methodology, enabling a better concept and technology exploration of aerospace architectures, is discussed in this section. The methodology is developed to address the challenges for the concept and technology selection of aerospace systems, as enumerated below:

Thesis Challenges:

1. Enable the exploration of a larger number of concepts
2. Enable the simultaneous concept and technology exploration
3. Facilitate the exploration of incompatible concepts and technologies
4. Structure the architecture selection process

In addition to these challenges, a set of evaluation criteria were defined in Section 2.7 for the benchmarking of the existing design methodologies. These criteria aim at 1) reducing the execution time, 2) reducing the convergence error, 3) enhancing the capability to explore quantitatively a matrix of alternatives with incompatibilities, and 4) maximizing the adaptability to new concepts and technologies. They are used to select the best choice of each component of the design process.

The overall methodology work-flow proposed for this thesis dissertation is thus presented in Figure 3.2. The proposed concept and technology down-selection methodology includes six steps. The first step represents the definition of the project requirements (objectives and mission). The second step determines a set of concepts and technologies, which then populate the morphological and incompatibility matrices in

Step 3. Step 4 consists of mapping the concepts to the design variables of the modeling and simulation environment. For example, rocket engines with various propellant types are mapped to the specific impulse (Isp) and the thrust-to-weight ratios (T/W) in the modeling environment. A different setting for the Isp and T/W is assigned to each rocket engine concept. This enables quantitative evaluation of each vehicle architecture, as mentioned by the Challenge 1 above.

Once the first four steps are completed, a design space exploration is performed. This phase consists of generating compatible solutions (Step 5) and sizing those solutions according to the modeling and simulation environment (Step 6). In other words, first synthesize and then size the concepts. At the end of the design space exploration, if the optimized architectures satisfy the requirements, they lead to a set of potential architectures that can be explored more thoroughly with other optimization techniques. If the optimized architectures do not satisfy the requirements, design engineers have to either modify the requirements or add new concepts or technologies to the design space. The next section develops some of the components required for the new design method proposed in Figure 3.2.

Before going to the next section, a look back at the traditional concept selection approach in Figure 3.1 or the traditional design process in Figure 2.6 can show the differences and similarities between this methodology and the traditional approach. On the one hand, the first four steps are indeed similar to the traditional approach. They consist of defining the problem, finding alternatives, and organizing the alternatives in morphological fields. On the other hand, the new method is different from the traditional approach in four major aspects. First, the technologies and concepts are simultaneously explored. Second, the method enables the quantitative design space exploration formed by a large pool of concepts and technologies. Third, incompatible concepts or technologies are taken into account in the design space exploration. Finally, the fourth difference resides in the mapping between the morphological matrix

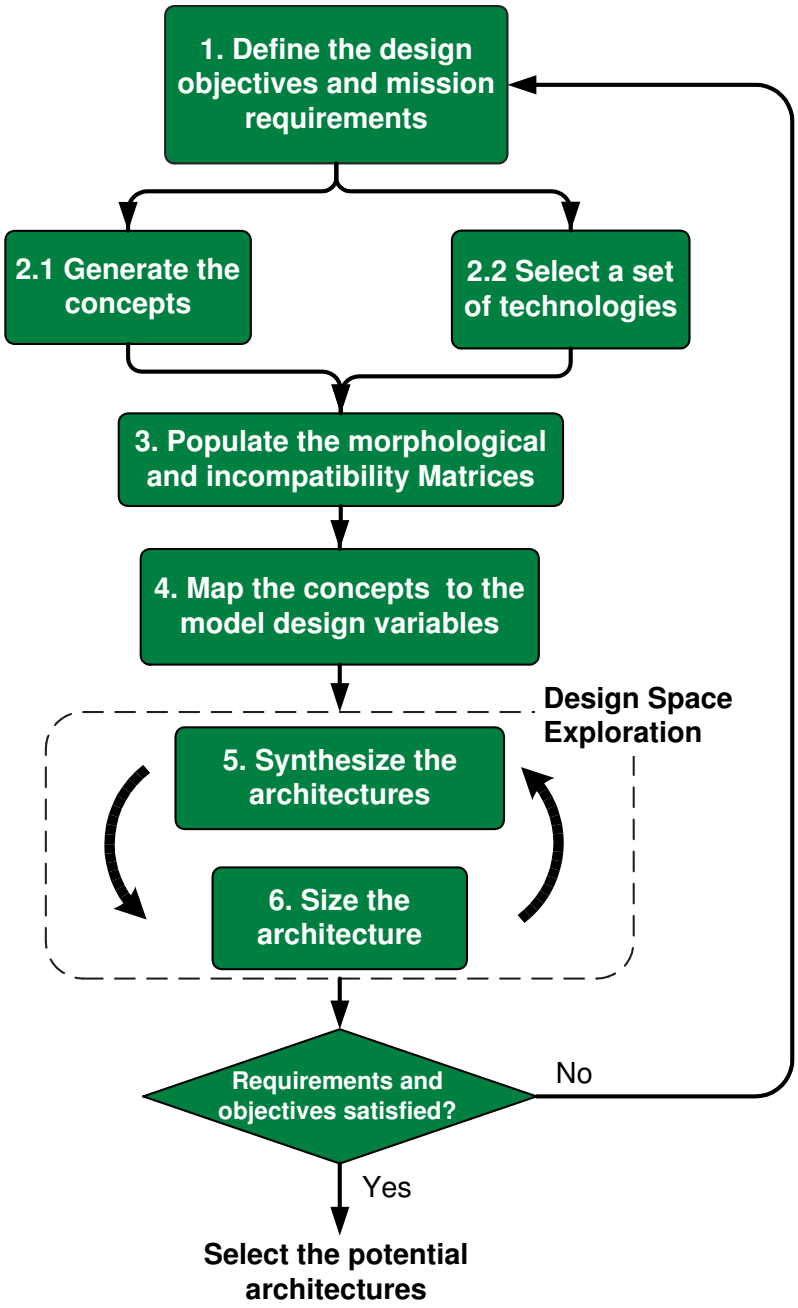


Figure 3.2: Proposed methodology for concept and technology exploration of aerospace vehicle architectures

and the modeling and simulation environment, which is introduced to assess quantitatively the performance of each architecture. The next section describes the different steps of the process in more detail.

3.3.2 Step 1: Defining the Problem

This first step is intuitive since it is similar to the traditional approach. Both the mission requirements and the design objectives are defined. For example, the mission requirements for a launch vehicle could be the orbit where the payload must be delivered, the payload weight, and the payload dimensions. This method assumes that the mission requirements are fixed. However, they can be changed after the design space exploration is completed. The goal of the design objectives are minimize the development cost, the unit production cost, or the gross liftoff weight. These are the figures of merit that help to compare the architectures.

3.3.3 Steps 2: Enabling Innovation

The second step of the design method presented in Figure 3.2 consists of generating alternatives that could satisfy the requirements and selecting a pool of technologies that could improve the architecture. This is the step which requires creativity to generate a large enough number of potential alternatives. It is a crucial step in a design project since the larger the number of alternatives the better the chances of generating a good solution.

3.3.4 Steps 3: Enabling the Simultaneous Exploration of Concepts and Technologies

It was noticed during the analysis of the Exploration Systems Architecture Study (see Section 2.1) that the technology selection was sometimes done separately from the concept selection. To solve this problem, the new design method proposes to insert the technologies in the morphological matrix in the same way as the concepts. Thus, the morphological matrix will have functions related to the concepts and other

functions related to the technologies. This approach will enable the simultaneous exploration of both the concepts and technologies. The morphological analysis, as described in Chapter 2, is an effective way to organize a design problem, and to develop new creative solutions. For this reason, it is implemented in the concept and technology selection process.

Does grouping the technologies and the concepts in the same morphological matrix makes sense from a design standpoint? Observing the characteristics of both the concepts and the technologies in a conceptual design perspective may provide an answer to this question. In fact, technologies and concepts are similar notions in conceptual design since they both represent discrete ways to model a design option. In addition, their level of performance on the vehicle can be mapped to the design variables used by the modeling and simulation environment. Thus, in the context of conceptual design, technologies and concepts are similar and can be grouped together in the same morphological matrix. This way of modeling the design space is the first step in enabling the simultaneous design space exploration of the concept and technologies. The rest of the approach is detailed in Section 3.3.6

3.3.5 Step 4: Mapping the Concepts and Technologies to the Modeling & Simulation Environment

The first three steps are qualitatively performed as they define the problem (Step 1), generate alternatives (Step 2), and organize those alternatives (Step 3). Step 4 is the bridge between the qualitative and the quantitative phases of the design process. As shown in Figure 3.3, Step 4 maps the discrete space formed by the concepts and technologies in the morphological matrix to the continuous space formed by the modeling and simulation input variables. This step is necessary to quantitatively assess the performance of each vehicle architecture generated during the concept exploration phase.

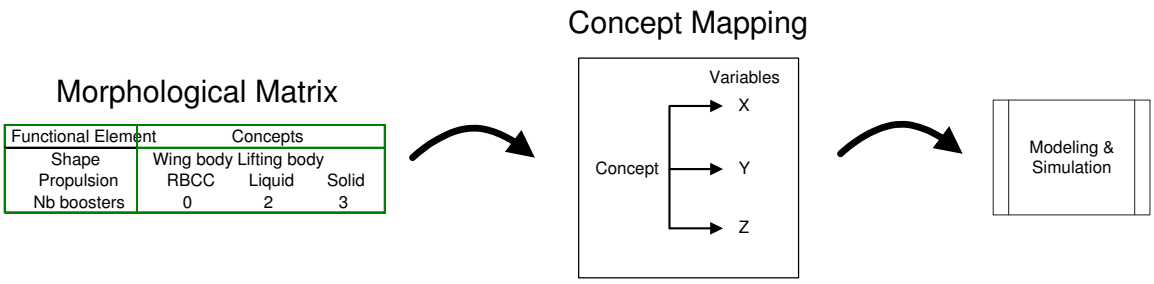


Figure 3.3: Concept to modeling and simulation mapping strategy

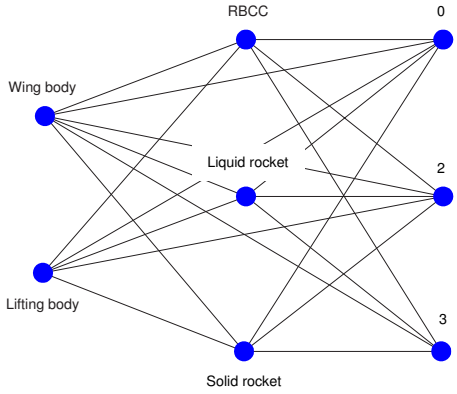
3.3.6 Step 5: Generating Compatible Architectures Using Graph Theory

It is stated in Step 3 that the morphological analysis is a effective way of organizing concepts and generating new solutions. However, the morphological matrix often requires human intervention to synthesize compatible concepts, because a large portion of the concepts in the matrix are incompatible. To carry out the overall design process, the morphological matrix must be computationally explored in such a way that only compatible architectures are transferred to the modeling and simulation environment. Thus, a new way of representing the morphological matrix is proposed herein to facilitate the design space exploration.

The modeling of the morphological matrix consists of using a graph, where the nodes represent the concepts and technologies available for a design, and the arcs represent a combination of two concepts. In other words, an arc represents a decision made by a design engineer to join two concepts together. The vehicle has a compatible solution when all the necessary systems in the morphological matrix have a selected concept or a technology. This process is illustrated in Figure 3.4 where a theoretical morphological matrix composed of three systems and three concepts or technologies is represented by a graph composed of nine nodes and 18 arcs (27 combinations).

Morphological Field	Concepts		
Vehicle type	Wing body	Lifting body	
Engine type	RBCC	Liquid rocket engine	Solid rocket engine
Number of boosters	0	2	3

(a) Morphological Matrix



(b) Morphological Graph

Figure 3.4: Transformation of the morphological matrix into a graph

The morphological matrix, however, does not indicate which concepts or technologies are incompatible. Therefore, an incompatibility matrix supplements the morphological matrix to take into account infeasible concept combinations. This incompatibility matrix is a square matrix that indicates which concepts are compatible by a value of 1, and which concepts are not compatible by a value of 0. Equation 3.1 shows the incompatibility matrix of the morphological matrix presented in Figure 3.4a. The incompatibility matrix, in this case, is the same as the adjacency matrix (explained in the next chapter), and it is symmetric. The values of 0 correspond in the matrix to the incompatibility between two concepts of the same morphological field. For instance, α_1, α_2 and α_3 cannot be selected in a same architecture since they are in the same morphological field and they accomplish the same function.

$$IM = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{3.1}$$

This approach facilitates the design space exploration of the morphological matrix. Besides, it can explore a morphological with incompatible alternatives, which represents a challenge in conceptual design. This notion will be later employed in the design space exploration phase since it allows the generation of compatible solutions. Furthermore, it enables the use of path-searching optimization algorithms, which should improve the efficiency of the combinatorial design space exploration. This is explained in the Section 3.3.8. Step 6 describes the sizing and synthesis approach used for evaluating the architecture performance.

3.3.7 Step 6: Sizing each Architecture Using the Sizing and Synthesis Environment

Once a set of compatible architectures is generated, the goal of Step 6 is to evaluate their performance by using a sizing and synthesis environment. The environment must have the capability of quickly modeling a large number of architectures.

In this thesis, the architecture exploration is applied to launch vehicle designs. Since no sizing and synthesis environment was available to the author with the required flexibility and capability of modeling a large variety of launch vehicle architectures, a new sizing and synthesis code was developed for sizing a large variety of

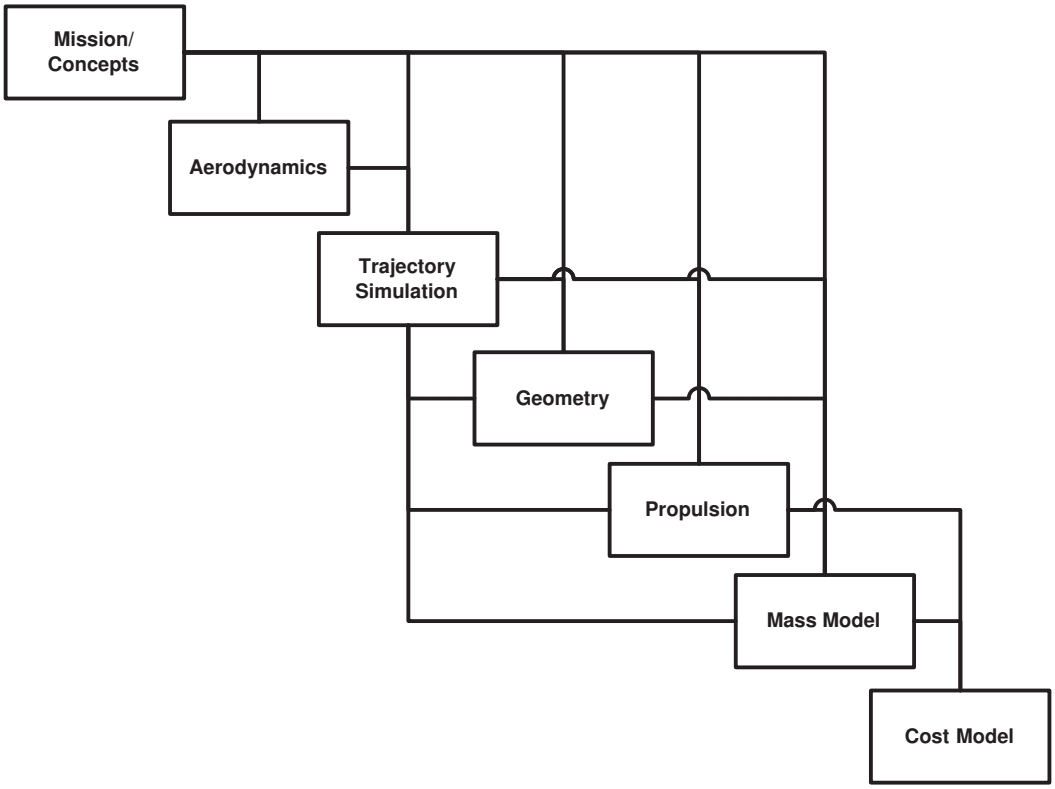


Figure 3.5: Design structure matrix Rapid Access to Space Analysis Code 3 (RASAC-3)

launch vehicle architectures. This code, presented in Chapter 6 is the Rapid Access-to-Space Analysis Code 3 (RASAC-3). It was first developed under the modeling philosophy that emphasizes on fast vehicle sizing, flexible trajectory simulation modeling, and simple cost evaluation. It was widely modified to assess launch vehicle architectures, hence receiving the name RASAC-3. As a first overview, the design structure matrix of RASAC-3, presented in Figure 3.5, shows the various disciplines involved. Chapter 6 describes this sizing and synthesis program in more detail.

3.3.8 Ant Colony Optimization for Design Space Exploration

Thus far, the six steps formulating the concept and technology exploration methodology are covered. The only item remaining is the actual design space exploration as shown in Figure 3.2. This section describes how this exploration is performed.

There are several combinatorial optimization algorithms used for a number of combinatorial problems [61, 79]. There is, however, a limited amount of combinatorial graph-based optimization algorithms that are directly applicable to the exploration of a morphological matrix. For example, Branch and Bound algorithms, and dynamic programming usually work well for the optimization of graphs such as the Traveling Salesman Problem (TSP). Unfortunately, the algorithms work by finding local solutions in the design space and bounding these local regions together to solve the bigger problem. Unfortunately, this approach cannot be used in the concept exploration of aerospace architectures because the algorithm cannot guarantee that those local solutions form a compatible concept. Moreover, other graph optimization algorithms such as physical programming do not work on exploring the morphological matrix because they require heuristic information to move over the graph. As an example, for the TSP, the distance between two cities is the heuristic information used to select the next move. This information is not available when selecting concepts because there is no way to determine the goodness of a solution by selecting a portion of the required concepts to form an architecture. In short, a vehicle alternative must be entirely constructed before assessing the performance. The only graph-based optimization algorithm known to the author that has the above characteristics is then Ant Colony Optimization (ACO), introduced in the next paragraph.

ACO is a relatively new optimization method developed from the observation of the efficient foraging behavior of ants in a colony. It is a path-searching algorithm that efficiently explores combinatorial problems because of the interaction between agents (ants) and a small level of disorder introduced by the randomness of ant behavior. This approach also shows a good convergence rate for the optimization of combinatorial problems such as the traveling salesman problem, job scheduling, and set covering [28]. The next chapter presents the theory relative to ACO for traditional combinatorial optimization problems. The application of ACO to architecture

selection is, however, different from traditional combinatorial problems. To explore morphological matrices, the algorithm follows the steps described below.

First, the initialization of the ACO algorithm must create a colony of m ants, where each ant represents an agent able to construct a concept, that will perform n iterations (tours). The number of ants must be large enough to ensure a global design space exploration and the number of iterations large enough to ensure a good convergence of the algorithm. The morphological graph and its adjacency matrix are created from the information contained in the morphological matrix and the incompatibilities between concepts and technologies. This creates an adjacency matrix that regroups the compatible concept and technology combinations.

The next step consists of allowing the artificial ants to construct compatible solutions (architectures) by traveling over the morphological graph. Each ant has to travel along the arcs of the graph with specified starting and ending nodes. This action is illustrated in Figure 3.6 where each path followed by an ant is represented by a different line type.

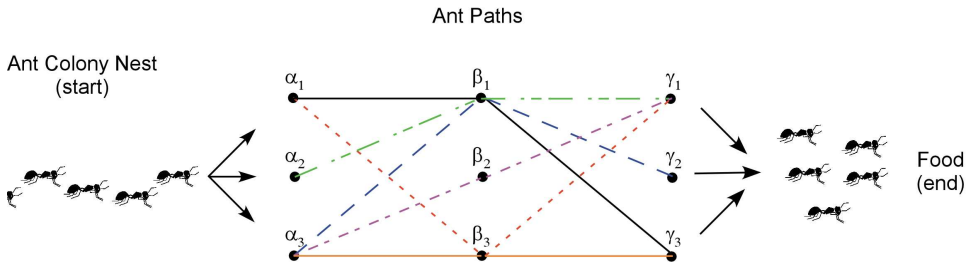


Figure 3.6: Ant Colony Optimization (ACO) applied to the morphological graph

The third step consists of evaluating the performance of the solution once the ants k of the tour m have arrived to the end of the graph. In other words, once all the architectures are constructed from compatible solutions, the sizing and synthesis code evaluates its performance. The results obtained from the modeling and simulation code determine the amount of pheromone[1] deposited on the path followed by the

[1]For a natural ant, a pheromone is a chemical deposited on the path followed to communicate

ant. It is advantageous to have a fast sizing and synthesis code that evaluates each architecture quickly and accurately. The amount of pheromone augmentation ($\Delta\tau$), usually proportional to the performance of the constructed solution, is then deposited on each arc forming the path according to Equation 3.2, where τ represents the pheromone level on the arcs. This steps is performed computationally by updating the pheromone matrix.

$$\tau_{ij}^{new} = \tau_{ij}^{old} + \Delta\tau^{k} \tag{3.2}$$

Finally, once the performance of each ant is calculated, a the pheromone is evaporated by a factor. The evaporation is done to let the algorithm eliminate bad solutions that have received a small amount of pheromone. This ensures a more global exploration of the design space. Figure 3.7 illustrates each steps mentioned above in a pseudo-code.

```
Set n = number of colony tours
Set m = number of ants in the colony
Set ρ = evaporation rate

For i = {1,..., n}

        For j = {1,...,m}

                Construct solution on the graph (1)
                Evaluate the solution (2)
                Deposit pheromones on the trail followed (3)

        End

        Evaporation (4)

End
```

Figure 3.7: Pseudo-code for the ACO algorithm

These optimization steps are performed iteratively until the algorithm has reached the total number of tours or convergence. The best solutions are recorded throughout the optimization process, and can be explored more thoroughly using a local search

with the other ants in the colony the paths that lead to good sources. In ACO, it is a notional quantity used to communicate the good combinations of concepts or technologies.

optimization algorithm. Chapter 4 details the theory of ACO in more detail.

CHAPTER IV

THEORY

> *"It is quite wrong to try founding a theory on observable magnitudes alone. In reality the opposite happens. It is the theory which decides what we can observe."* [Heisenberg, 1971]

This chapter presents the underlying theory used to develop the new concept and technology selection process detailed in Chapter 3. It covers the theory behind architecture design, the morphological analysis, graph theory, and Ant Colony Optimization (ACO).

4.1 System Design and Architecture Selection

Thus far, the concept of architecture selection has come up quite often in the discussion. For this reason, this section defines some concepts behind architecture selection and presents in what context it is used and is helpful.

The selection of architectures, although a broad topic of investigation, is a subtask of an even broader process: system design. In the International Council on Systems Engineering (INCOSE) handbook, architecture selection is one of a series of three iterative steps. The other two steps are the requirements definition and the functional analysis. A representation of the system design iteration loop is presented in Figure 4.1, showing where the architecture exploration is located in the design process.

Architecture design is a complex task and its meaning is subject to interpretation across the engineering industry. The INCOSE defines it as the "*selection of the types of system elements, their characteristics, and their arrangement*" [38]. There

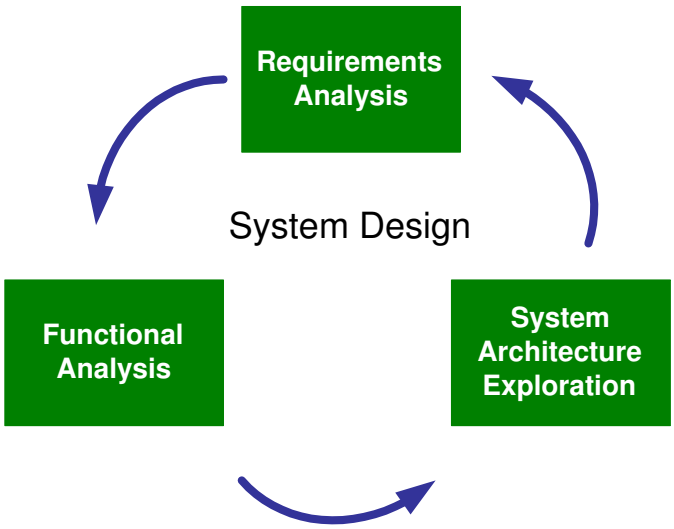


Figure 4.1: System design process. Adapted from the INCOSE handbook [38]

exist different types of architectures for a system: the functional architecture, the physical architecture, the technical architecture, and the operational architecture. Morton [66] defines these four types of architectures in the following way. The functional architecture refers to a hierarchy of functions that the system must perform for the given requirements. The physical architecture relates the subsystem and components for a system. The technical architecture refers to the interface between components. Finally the operational architecture relates to the way the elements operate and interact to accomplish its goal. This thesis mainly explores the physical and technical architectures for a launch vehicle, as the other two architectures are fixed, as they are usually performed before the technical and physical architecture selection.

The exploration of system architectures is important in system design. First, it usually improves the tracking of the requirements throughout the conceptual design phase thus putting the customer in the front row of the design process. Second, it helps in obtaining better solutions because it enables the generation of a larger number of solutions, and it enables the creation of more innovative alternatives.

Although different techniques can be used to generate architecture alternatives, this thesis uses the morphological matrix, a subset of the morphological analysis. The description relative to the morphological analysis is detailed in the next section.

4.1.1 Systems Engineering and Launch Vehicles

With the emergence of complex vehicle systems, Systems Engineering is taking more importance in the aerospace industry. To this extent, several firms in the aerospace industry have adopted a systematic approach to perform their vehicle design. Also, INCOSE was created to improve the dissemination of knowledge and ensure better design of systems.

Before going into much details, some terms must be defined. First, according to the INCOSE, a system is defined as an "integrated set of elements that accomplish a defined objective." Similarly, Jackson defines the term system as an"interacting combination of elements, viewed in relation to function". [52] Thus, a system can be viewed as any set of components integrated together to perform a specific task, or function. A system for a particular engineer may constitute a sub-system for another. For example, an engine designer may view the engine as a system and the engine components as subsystems. The aircraft manufacturer, however, views the engine as a system and its aircraft as a system. Hierarchy is thus an inherent property of a system.

Second, Systems Engineering represents the creation of systems in order to accomplish as specific purpose. Nowadays, the focus is not on product engineering but systems engineering since the complexity of engineering systems require a more global and integrated design approach. Systems Engineering is defined more specifically as the *"interdisciplinary approach and means to enable the realization of successful systems."* [38] This statement highlights the importance of systems design with an integrated multidisciplinary approach.

Over the years, systems engineering has emerged as the process followed to ensure a satisfaction of all the customer's needs. The eight-step process presented in Table 4.1, proposed in the INCOSE Handbook [38], illustrates the tasks required to facilitate the work of systems engineers.

Table 4.1: INCOSE systems engineering process [38]

Step	Description
1	Define the system objectives
2	Define the functionality
3	Establish the performance requirements
4	Evolve the design and operations concepts
5	Select a baseline
6	Verify that the baseline meets the requirements
7	Validate that the baseline satisfies the user
8	Iterate the process through lower level analysis

As mentioned in the INCOSE Handbook "a broad search strategy that explores the whole design space with some reasonable sampling density is needed" [38]. This problem is directly related to steps 4, 5, and 6 of the above process, which justifies the focus of this thesis on these same steps. In order to perform enable the accomplishment of these three steps, a system architecture must be defined and iterated, as explained in the next paragraph.

The iteration of the architecture is referred more or less to the exploration of different design options, or vehicle architectures in the case of launch vehicles. To get to this point, the functional architecture and the physical architecture of the product must be brought forward in the design process. A functional architecture is a hierarchical representation of the function that a product must accomplish. A physical architecture is a hierarchical representation of the different components of a system. System architecture synthesis is defined as "the selection of the types of system elements, their characteristics, and their arrangements".[38] This process requires the exploration over a large, non-linear design space, which also has several local minima. This characteristic of the design space thus justifies a system design

exploration algorithm that performs efficiently on these types of design spaces.

4.2 Morphological Analysis

The morphological matrix is one of the central concepts of the architecture selection process proposed. Developed by Zwicky [108], the morphological matrix is an enabler of a innovative idea generation technique he developed: the morphological analysis. The approach consists of decomposing a complex problem into categories, and generating alternatives for those categories. New concepts can be generated by joining compatible alternatives from each category. The goal of this method is thus to evaluate the relationships in non-quantifiable multidimensional problems.

The method can be summarized in five different steps: [109]

Step 1: Problem formulation

Step 2: Determination of the important factors influencing the problem or the solution

Step 3: Enumeration of the solutions, and construction of the morphological matrix by regrouping the solutions in a set of parameters

Step 4: Analysis of all the combinations of solutions in the morphological analysis

Step 5: Selection of the optimal solution

Morphological analysis enables a better organization of ideas and more creative thinking towards new solutions. It also facilitates the visualization of the problem and of the possible solutions.

One of the major tools developed by Zwicky for the application of the morphological analysis is the morphological matrix (also called morphological field). This tool was created in order to facilitate the manipulation of information and solutions relative to the problem. The tool allows the representation of complex multi-dimensional combinatorial sets in a much easier fashion, enabling the user to better understand

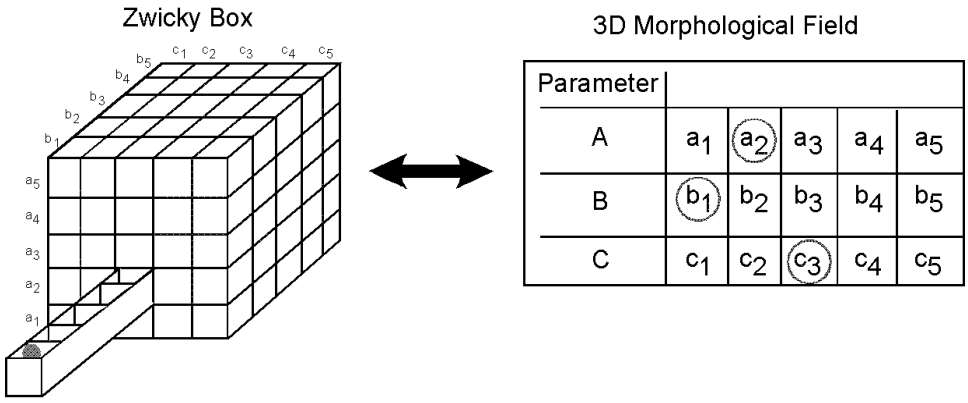


Figure 4.2: Graphical representation of the morphological matrix. Adapted from Ritchey [80]

the problem and its alternatives. For example, the concepts of an aircraft are more manageable when grouped into subsystems than when all considered at the same time. For an aircraft divided into three systems with five concepts per systems, this would represent 5^3 or 125 alternatives, which is more difficult to manage than if the concepts were grouped in a morphological matrix, as shown in Figure 4.2. This figure in fact illustrates the advantages of representing a 3-dimensional configuration space having five options per dimension by a morphological matrix *versus* a cartesian set of coordinates (Zwicky Box). The circled options in the morphological field (Figure 4.2b) are equivalent to the dot location in the Zwicky box (Figure 4.2a).

Zwicky applied his method to astrophysics, space travel, propellants, and propulsive engines. As an illustration, Zwicky used his method for the generation of new propulsive engine concepts using chemical energy. Six different factors were considered for the study; i.e., 1) environment in which the engine is moving, 2) type of motion of the propellant, 3) physical state of the propellant, 4) type of thrust augmentation, 5) type of ignition, and 6) type of operating cycle. For each factor, different options were imagined, as they are represented in Table 4.2.

This analysis of the jet-engine morphological matrix revealed more than 500 new engine types, a number of which were granted patents. [109] It thus shows how useful

Table 4.2: Example of the morphological analysis applied to the jet engine propulsion [109]

Factor	Concepts		
Outside environment*	Vacuum	Atmosphere	Water
Type of propellant motion	Translatory	Oscillatory	Rotatory
Physical state of propellant	Gaseous	Solid	Liquid
Type of thrust augmentation	None	Internal	External
Type of ignition	Self igniting	External ignition	
Sequence of operation	Continuous	Intermittent	

* Referred as the environment in which the body moves

the method is in structuring the design solution and enabling creative thinking.

For this thesis, the morphological analysis is used to represent the concepts and the technology alternatives for an aerospace architecture. The standard way of using the morphological matrix, however, does not enable the quantitative analysis of the alternatives. For this thesis, however, the method proposed in the previous chapter enables the quantitative exploration of the morphological matrix. Consequently, this new capability should enhance even more the benefits of the morphological analysis. The next chapter shows how the morphological matrix is modeled to enable its quantitative exploration.

4.3 Graph Theory

As detailed in Chapter 3, graph theory is of one of the keystone concepts that enables the development of the new concept and technology selection process. This section thus defines some of the terms used in graph theory.

4.3.1 Definitions

Graph theory is an old field of mathematics, yet powerful because it can easily represent the relationships between different factors. By definition, a graph $G = (V, E)$ is a set of vertices (nodes) interconnected through edges (arcs). The number of vertices of a graph (G) is called the *order* of G and the number of edges the *size* of G. Graphs

can constrain the movement from two nodes to a specific direction (*directed*) or not (*undirected*). Moreover, a *path* in a graph can be defined by the edges followed to go from a starting node to an ending node. A *circuit* in a graph is thus a path that ends where it started. Figure 4.3 shows three graph examples, that is, two undirected graphs (a) and (b), and one directed graph (c).

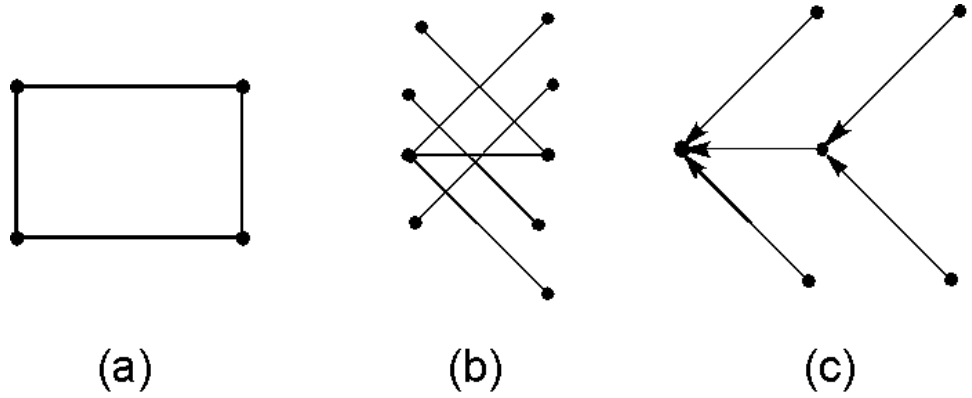


Figure 4.3: Three (3) graph examples

4.3.2 Applications

Despite their simplicity, graphs have numerous and diversified applications. For example, they are used widely for the modeling of printed circuits, fleet routing, molecular chemistry, and job scheduling. [41] The latter is not necessarily obvious but let us imagine that four jobs j have to be performed by two workers w. This case can be represented by the graph in Figure 4.4, which demonstrates the usefulness of a graph in visualizing a problem and its solutions.

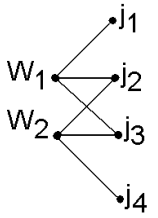


Figure 4.4: Job scheduling example

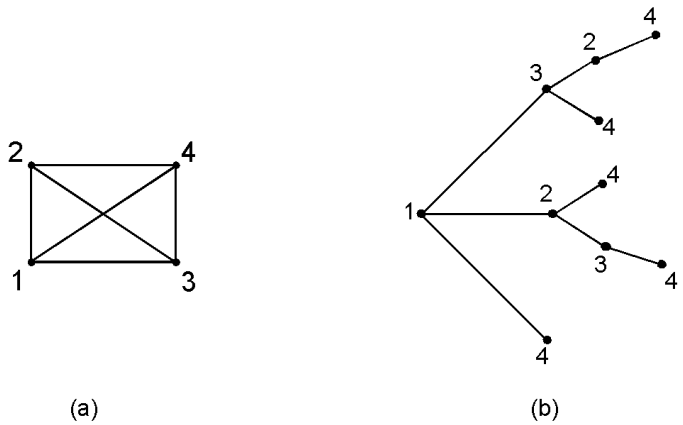


Figure 4.5: Equivalency between a graph (a) and a spanning tree (b)

4.3.3 Trees

A tree is another type of graph. It is defined as a directed graph with no circuit. [98] A tree usually starts from a common node and then spans up into branches. Trees have found a variety of applications because of their ability to illustrate hierarchical problems such as the hierarchy of a company. In addition, they can easily decompose a problem or the factors that influence a phenomenon into simpler elements. Interestingly, regular graphs can also be represented by an equivalent tree, called a *spanning tree*, which makes the graph's analysis easier to perform. Figure 4.5 represents a graph example and its equivalent spanning tree for all the possible paths that depart from node (1) and end to node (4).

4.3.4 Adjacency Matrix

In order to facilitate the handling of graphs with computers, graphs can be represented in a matrix form called the *adjacency matrix*. This type of matrix can be defined as an n by n matrix with a 1 at row i and column j if the nodes i and j are connected, where n represents the number of nodes of the graph. For undirected graphs, the nodes are symmetrical, which can simplify their manipulation. The matrix below is the adjacency matrix of the graph displayed in Figure 4.5a.

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \tag{4.1}$$

Finally, a common denominator between graph theory and the morphological matrix is that they both have the goal to represent the relationship between the factors involved in the description of a problem. This observation was made during a search for architecture selection alternatives, and it led to the modeling of the morphological matrix by a morphological graph. This connection between the two methods facilitates considerably the design space exploration for two main reasons. First, a morphological graph facilitates the generation of compatible solutions. Indeed, the major problem behind automatic generation of alternatives from the morphological matrix was the generation of incompatible solutions, which does not happen when a graph is used to generate compatible solutions. The second reason is that a morphological graph enables the application of a graph-based stochastic optimization algorithm, which can search the design space more efficiently than traditional stochastic optimization methods (e.g., SA, GA). This subject is treated in the next chapter.

4.4 A Graph-Based Optimization: Ant Colony Optimization

Ant Colony Optimization (ACO) was introduced in Chapter 3 as an enabler for the new concept and technology selection process shown in Figure 3.2. ACO is a stochastic optimization approach that works well on the optimization of combinatorial problems modeled by graphs. This method is selected as the optimization algorithm of choice for exploring the morphological matrix, which is why it is detailed in the next sections.

4.4.1 How Natural Ants Work?

ACO is one of the optimization algorithms that finds its effectiveness in reproducing the behavior of biological organisms. In this case, it is the behavior of foraging ants. The behavior of natural ants is in fact impressive. These colonies of insects interact according to structured social organizations. For example, each ant in the colony accomplishes a specific function. Some ants transport food, others nurse larvae, and some others are foraging to find good food sources close to the nest. The behavior of foraging ants in search of food is a spectacular example of the level of social interaction between these insects[27].

Foraging ants, yet almost blind, communicate between each other through a chemical substance they produce called pheromone. Goss *et* al. [37] shows by a simple experiment that ants tend to use the shortest path toward a food source when there is a higher concentration of pheromone on the path. The authors set up a double bridge experiment, as shown in Figure 4.6, where ants were initially placed in the nest with two paths leading to a food source. In the first case the branches are of equal lengths whereas in the second case, the branches were of different lengths. The experiment shows that ants will likely follow the shortest path to a food source because they communicate good paths to each others with pheromones. Figure 4.7 shows this behavior, by displaying the number of experiments where the group of ants followed the shortest path. This experiment thus confirmed that foraging ants cooperate by means of some simple communication [17].

The double bridge experiment does not show the behavior of ants in the case of multiple food sources. Addressing this issue, Beckers *et* al. [4] shows that the amount of pheromone left on the trail followed by one ant is proportional to the richness of the food source when multiple sources of food are available. This interesting ant colony behavior constitutes the foundation of Ant Colony Optimization, discussed in the next section.

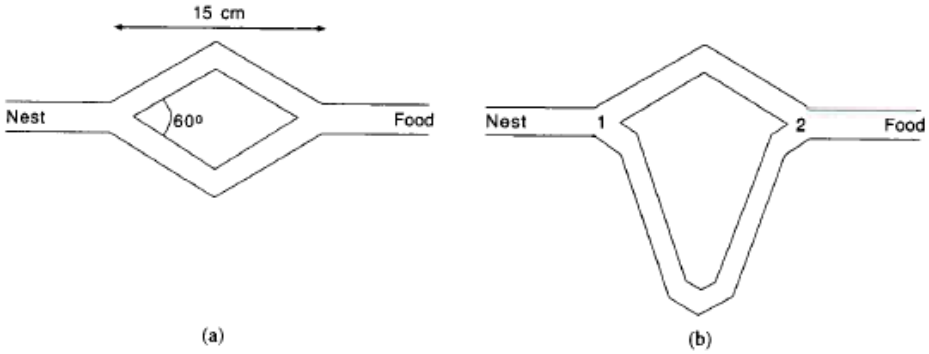


Figure 4.6: Experimental setup of the double bridge experiment. Branches lengths are (a) equal, and (b) different. Ref.[28]

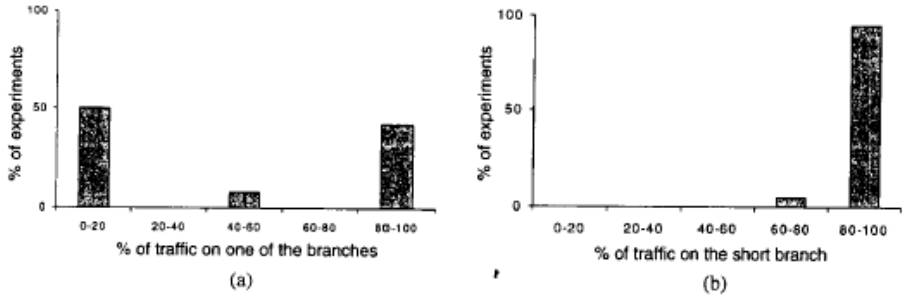


Figure 4.7: Experimental results of the double bridge experiment with branches lengths (a) equal, and (b) different. Ref.[28]

4.4.2 How Artificial Ants Work in ACO?

Similar to natural ants, ACO is an iterative path-searching optimization algorithm that reproduces computationally the interaction of ants in their colony in a search for the optimum of a function. It was developed by Dorigo[29]. Practically, the method consists of creating a colony of interacting ants that search over graph. Ants in the colony then construct solutions and deposit pheromones on the paths leading to good solutions. At the end of the iteration, the pheromones deposited on the paths are evaporated to simulate the evaporation occurring in nature. The ACO procedure can be summarized in three steps, as shown below. Each step is detailed in the following sections.

1. Solution construction through probabilistic path searching method
2. Pheromone deposition on the path based on the value of the objective function
3. Pheromone evaporation

4.4.2.1 Solution Construction

The first step that ACO performs is the solution construction. This step is the process by which an ant moves over a graph, node by node, until it forms a complete solution of the problem represented by the graph. At any given node, each ant selects one of the available arcs with a probability computed from Equation 4.2,

$$p_{ij}^{k} = \begin{cases} \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{l \in N_i^k} \tau_{il}^{\alpha} \cdot \eta_{il}^{\beta}} & if j \in N_i^k \\ 0 & if j \notin N_i^k; \end{cases} \tag{4.2}$$

where p_{ij}^{k} is the probability for an ant k located at the node i to go to node j, and τ_{ij} is the pheromone concentration of arc (i, j). In addition, p represents the probability for an ant k located at the node i to go to node j, τ_{ij} is the pheromone concentration of arc (i, j), N_i^k represents the neighborhood (i.e., feasible moves)of an ant k at node i, l the feasible nodes for an ant k, and α the relative importance of pheromone level

over the heuristic information (β) contained in η_{ij}. Equation 4.2 basically states that the probability of moving to a node is a function of the pheromone amount on the arc and some other preliminary information (β) on the goodness of the move. For the TSP problem (see Section 6.74) the preliminary information is the distance between two nodes. However, for the exploration of the morphological graph $\beta = 0$ because no heuristic information is available to dictate the goodness of a move until the entire solution is constructed and evaluated.

4.4.2.2 Pheromone Trails Update

The next step in ACO consists of updating the pheromone concentration on the trails followed by each ant in the colony. The procedure consists of evaluating the performance of each solution constructed by the path and then of depositing a certain pheromone quantity on each of the arcs forming the ant path. The quantity of pheromone deposited ($\Delta\tau^k$) is proportional to the goodness of the solution. For example, if the cost per flight is the figure of merit, then Vehicle A, which has half the development cost of Vehicle B, would receive twice the amount of pheromone on its arcs. This procedure updates the overall pheromone matrix, which tracks all the pheromone change on a graph. A mathematical representation of the procedure is presented in Equation 4.3, where τ is the pheromone level on the arc (i, j), are the nodes of the graph, $\Delta\tau$ is the pheromone augmentation, and k is the pointer to a specific ant.

$$\tau_{ij}^{new} = \tau_{ij}^{old} + \Delta\tau^k \tag{4.3}$$

A pheromone matrix is used to update the amount of pheromone (τ) on the graph. This matrix is an n by n symmetric matrix, where n represents the number of nodes. Equation 4.4 shows an example of a pheromone matrix. The goal of the matrix is to keep track of the pheromone concentration on the arcs of the graph to guide the

exploration of artificial ants.

$$M_P = \begin{bmatrix} \tau_{1,1} & \cdots & \tau_{1,n} \\ & \ddots & \\ \vdots & \tau_{i,j} & \vdots \\ & & \ddots \\ \tau_{n,1} & \cdots & \tau_{n,n} \end{bmatrix} \tag{4.4}$$

As mentioned in the paragraph above, the value of $\Delta\tau^k$ is proportional to the goodness of the objective function. For example, for an objective function that is minimized, Equation 4.5 is used whereas for a problem where the objective function is maximized, Equation 4.6 is used. There is a different value of the objective function for each F^k, where k is an ant in the colony, and F_{ref} represents a referenced value for the objective function.

$$\Delta\tau^k = \frac{F_{ref}}{F^k} \quad (Minimized) \tag{4.5}$$

$$\Delta\tau^k = \frac{F^k}{F_{ref}} \quad (Maximized) \tag{4.6}$$

The effect of the variable F_{ref} is important on the convergence of the ACO because it changes the quantity of pheromone deposition on the graph, hence the value of the pheromone matrix. A large value of F_{ref} corresponds to low pheromone deposition by an ant, thus leaving more chances for other ants to choose different arcs. In the opposite case, for low values of F_{ref}, a larger quantity of pheromone is deposited on the graph and other ants tend to follow the paths that were already followed. In other words, the tendency in ACO to explore the graph more thoroughly is directly proportional to the value of F_{ref}.

4.4.2.3 Pheromone Trail Evaporation

Pheromone evaporation consists of modeling the physical phenomenon by which the pheromone evaporates over time, which results in a decrease of the pheromone level in the pheromone matrix. Although it has little effect on real ants searching for food sources [28], the pheromone evaporation can have a significant impact on the ACO performance. Actually, evaporation is a necessary mechanism that avoids premature convergence by eliminating the poor solutions constructed in the past. [28]

Similar to the deposition procedure, the evaporation process is modeled in ACO. It is modeled by reducing the pheromone level by a ratio ρ at the end of each iteration. This procedure is expressed by Equation 4.7,

$$\tau_{ij}^{new} = (1 - \rho)\, \tau_{ij}^{old}, \forall (i, j) \in G \tag{4.7}$$

where ρ represents the evaporation rate, (i, j) an arc going from node i to node j, and G the graph.

In addition to making the algorithm forget the poor solutions from the past, the evaporation rate is also an important factor for the convergence and quality of the solution found by ACO algorithms. Dorigo and Stutzle [28] show that an optimal setting for the evaporation rate will lead to better performance of the algorithm. Along the same lines, values of the evaporation rate that are too low or too high result in the convergence of sub-optimal paths.

4.4.3 Final Comments on ACO

As shown in the previous sections, ACO is a powerful method to explore combinatorial optimization problems that can be represented by graphs. In addition to its effectiveness, ACO is a meta-heuristic optimization algorithm. That is, it can be used to optimize a large variety of problems. For example, when ACO is used to solve the TSP, vehicle routing, quadratic assignment, job scheduling, graph coloring,

and knapsack problems, a better convergence rate is typically seen when compared to other traditional optimization algorithms such as GA or SA. [28]

The procedure to apply ACO to a combinatorial optimization is relatively simple. The five steps enumerated below must be followed.

1. Construct solution paths
2. Evaluate the objective function
3. Update pheromone accumulation
4. Update pheromone evaporation

As proposed by Dorigo and Stutzle [28] a daemon activity can also be added to the algorithm to improve the quality of the solution. A daemon activity is a local search of the design space that cannot be performed directly by the ACO algorithm. For exploring the morphological matrix this procedure could be applied to optimize locally the solutions that are found promising by the algorithm. For instance, a daemon activity could be used to optimize the design parameters of a rocket engine once ACO has found that this concept forms a good architecture. Finally, despite its origin for combinatorial optimization problems, ACO has shown its effectiveness in optimizing multi-objective problems [26], as well as optimization under uncertainty. [39]

4.5 Traditional Discrete Optimization Approaches

Despite the new interests in Ant Colony Optimization, there are other combinatorial optimization techniques that enable the exploration of discrete design spaces. Two of the best known techniques are Genetic Algorithm (GA), and Simulated Annealing (SA). These two optimization algorithms will serve as benchmarks for the launch vehicle technology and concept exploration problem, presented in Chapter 7, as well as the technology selection problem, presented in Appendix D. For this reason, a brief summary of both optimization algorithms is presented in the following sections.

4.5.1 Genetic Algorithm

Genetic Algorithm (GA) is a stochastic optimization method based on Darwin's principle of natural selection. Its developer, John Holland at the University of Michigan, came up with the idea in the early 1970's of imitating the selection process of biological systems. [67] Its application is fairly simple. First the design variables are coded in a series of binary strings (genes). This coding generates a population of chromosomes. The optimization process consists of generating new members of the population by performing three genetic operations: 1) reproduction, 2)crossover, and 3)mutation. These operations consist of exchanging genetic information between the various members of the initial population. Finally, once the three genetic operations are done, the algorithm keeps the members of the population with the best fitness values and the process is repeated until the algorithm converges to a value.

GAs are used in both continuous and discrete solution spaces. The focus here is on discrete design space exploration because of the characteristics of architecture selection problems. A drawback to using GA is when exploring large design spaces, the algorithm usually requires more function calls than some other optimization algorithms to get to the optimal solution of a design space. In addition, GA sometimes converges slower than other optimization algorithms, and this can lead to sub-optimal solutions for certain problems if the number of iterations is not large enough. Despite its few drawbacks, GA is a powerful optimization algorithm. Its main strength is its capability of exploring a problem space more globally than many other optimization algorithms. The algorithm tends to retain good solutions through reproduction and to explore a larger diversity of solutions through crossover and mutation.[67] Finally, when a GA is paired to a local searching method, it can significantly improve the convergence rate and the solution.[106]

Several variations of the basic GA, as initially proposed by John Holland, have demonstrated significant improvements on the design space exploration of aerospace

vehicles. Moreover, its application on other types of vehicles [5, 43, 53, 67] show the good potential of GAs.

4.5.2 Simulated Annealing

Simulated Annealing (SA), such as GA, is a stochastic optimization method. SA optimizes a solution space by simulating the physical rearrangement of molecular energy levels that occur during the cooling process (annealing) of a metal. During a SA optimization, the objective function acts as the molecular energy, the variable settings as the molecular state, and the change in configuration as the change in molecular energy states of the material [86]. The specifics behind the application of SA to an optimization problem is described in the following paragraph.

During the design space exploration, SA algorithms start with population members randomly generated. This initial population represents an initial set of variable settings. A cooling process is simulated by allowing the population members to jump to different energy states thus enabling the generation of new solutions. If a new variable setting has a lower energy level (i.e., objective function), the change is accepted with a probability of one. Otherwise, that is, if the energy is higher, the algorithm accepts the change with a probability equal to Equation 4.8. The cooling process is modeled by a diminishing temperature schedule. Each iteration, represented by a specific temperature, is executed by allowing the previous population members to change configuration. This process ends once the final temperature is reached and the population does not change.

$$P(configuration = x) = 1 - e^{\frac{-\Delta E}{T}} \tag{4.8}$$

$$\Delta E = E\left(y\right)_i - E\left(y\right)_{i-1} \tag{4.9}$$

$$\tag{4.10}$$

where,

T = Temperature
x = New optimization vector
y = Set of old configuration
i = Iteration number
E = Energy

SA is not as efficient as GA in optimizing solution spaces. This poorer performance is because of the lack of interaction between the population members. In genetic algorithms the crossover and mutation genetic operations exchange the genetic information (solution settings) between the population members, which results in exchanging genes (solution parameters) between the population. By contrast, SA starts from a randomly generated configuration and changes the solution by hovering randomly in the solution space. This approach results in a less efficient optimization algorithm. Its adaptability to the optimization problem though may justify its use in discrete optimization problems. For example, SA can search more over a graph than GA.

Now that the concepts used in the design methodology are laid out, the next chapter describes how these concepts are applied to the development of a concept and technology selection methodology.

CHAPTER V

FORMULATION: CONCEPT AND TECHNOLOGY EXPLORATION ENVIRONMENT

This chapter discusses the formulation of the concept and technology exploration environment. This formulation is based on the methodology described in Chapter 3, and includes the concepts presented in Chapter 4. The following six sections describe each step of the design process, as presented in Figure 3.2.

5.1 Step 1: Define the Requirements

The first step of the technology and concept exploration methodology consists of defining the design requirements. This step, although intuitive, is critical because the requirements can have an impact on the selection of the best vehicle architecture, as illustrated in Chapter 2, in the context of the Space Shuttle design overview.

The first element in building the design requirements is to determine the design objectives. For this thesis, the design objectives are integrated into an overall evaluation criteria to facilitate the design space exploration. For example, the cost per pound of payload to orbit is a metric for evaluating different launch vehicle architectures.

The mission is defined in this first step, in addition to the design objectives. For example, the launch site and the type of orbit are examples of mission requirements that can be used.

Step 1 is easy to implement as it is similar to traditional conceptual design approaches, as presented in Chapter 2. Step 2 and 3, are different and they are detailed

in the next section.

5.2 Steps 2 and 3: Generate and Organize Alternatives

The next steps consist of generating alternatives (Step 2) and organizing them in the morphological matrix (Step 3). For the generation of alternatives, there is a large variety of concept generation techniques that can be used such as brainstorming, brainwriting, analogy, and cross-fertilization [33]. Any technique, or combination of techniques, can be used to perform this task. For example, one may consider the following options to design a single stage vehicle to go to orbit:

- Wing body, lifting body
- Liquid and solid rocket engines
- Rocket-Based Combined Cycle propulsion (RBCC)
- 0, 2, or 3 boosters

Once the concepts are generated, they are organized in the morphological matrix. This task is done by first defining a set of morphological fields that are placed in the first column of the morphological matrix. For a large architecture, the two levels of morphological fields should be created. For instance, the launch vehicle concepts can be grouped in the first and second stages categories. This is not the case for the notional example presented but the concepts enumerated above but it is addressed in the next chapter. In fact, this notional problem requires only three morphological fields: 1) vehicle type, 2) engine type, and 3) boosters. The concepts generated earlier can then populate the morphological matrix by grouping them in the three morphological fields, leading to the morphological matrix presented in Table 5.1. Thus, the first morphological field, vehicle type, has two alternatives: wing body and lifting body. The second one has three alternatives: Rocket-Based Combined Cycle, liquid rocket engine, and solid rocket engine. Obviously, a single stage with

solid propellant is unlikely to have good performance for an access-to-space mission but the morphological matrix does not differentiate yet between these technical issues. Finally the last morphological field includes the number of boosters where no boosters, 2 and 3 boosters are the alternatives available. The next section described how the morphological matrix is linked to the modeling and simulation environment to enable the quantitative exploration of vehicle alternatives.

Table 5.1: Notional morphological matrix

Morph. field	**Concepts**		
Vehicle type	Wing body	Lifting body	
Engine Type	RBCC	Liquid rocket engine	Solid rocket engine
Number of boosters	0	2	3

5.3 Step: 4: Mapping the Concepts to the Modeling & Simulation Environment

One challenge undertaken in this thesis is to automate the exploration of the morphological matrix, and 2) to generate architectures from a pool of concepts with incompatibilities. Section 5.3.1 addresses the first challenge whereas Section 5.3.2 addresses the second one.

5.3.1 Task 1: Transforming the Morphological Matrix into a Concept String

The goal of this first task is to automatically generate architectures from the morphological matrix. This task requires three operations: 1) encoding the morphological matrix, 2) mapping each concept in binary string, and 3) generating concepts from the binary string. Each of these three operations are explained in the next paragraph.

The first operation consists of encoding the morphological matrix. This is a task where the morphological matrix, initially created in a spreadsheet format is transferred into a text-based. This approach allows one to modify the morphological matrix if more concepts must be added. The text-based morphological matrix is then

read by the concept exploration environment and stored in cell array in the Matlab environment. A summary of these three steps is is depicted in Figure 5.1.

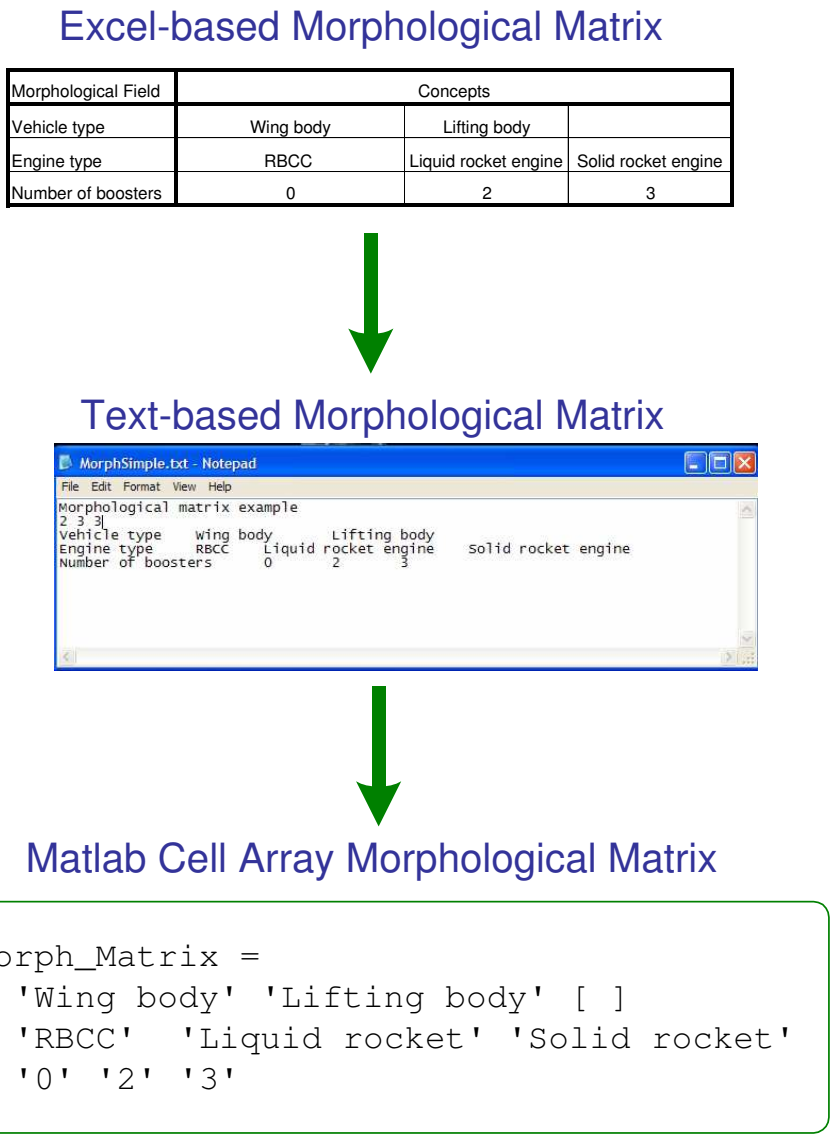


Figure 5.1: Morphological matrix modeling

Once the morphological matrix is loaded in Matlab, the second operation consists of assigning a digit to each concept. For the morphological matrix defined in Table 5.1, one would assign eight digits for each of the eight concepts, as shown in Table 5.2. This digit assignment is similar to GA or in fact any other stochastic optimization technique. This approach is flexible as the number digit assignment can

Table 5.2: Notional morphological matrix digit assignment

Morphological field	Concept	Assigned digit
Vehicle type	Wing body	1
	Lifting body	2
Engine type	RBCC	3
	Liquid rocket engine	4
	Solid rocket engine	5
Number of boosters	0	6
	2	7
	3	8

quickly be updated if more concepts are added into the morphological matrix. This numbering approach is useful because it helps the design engineer to add new alternatives without requiring much computational complexity. This usually enhances creativity and improves the final solution. [93]

The last operation consists of creating a concept string that reflects the different choices available in the morphological matrix. The concept string (S_C) can be represented by Equation 5.1, where β_i is a bit allocated to concept i. Each bit β_i has a value of 1 if the concept is part of the solution or a value of 0 otherwise.

$$S_C = \left[\begin{array}{ccccccc} \beta_1, & ..., & \beta_i, & ..., & \beta_n \end{array} \right] \tag{5.1}$$

For example, a wing body, propelled by liquid rocket engines and two solid rocket boosters architecture is defined by the concept string presented in Equation 5.2, where each digit of the concept string is defined in Table 5.2. The concept string's length is always equal to the number of concepts in the morphological matrix. Moreover, a concept string is associated to each population member, thus for each architecture created. It is analogous to the design vector in an optimization problem since it represents a state in the problem space explored.

$$S_C = \left[\begin{array}{cccccccc} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{array} \right] \tag{5.2}$$

This way of constructing design alternatives is fast, and it enables the automatic construction of concepts. The only issue of using this approach is that it does not guarantee the generation of compatible solutions. To resolve this issue, the incompatibility matrix is linked to the concept generation process, as discussed in the next section.

5.3.2 Task 2: Modeling the Incompatibility and Adjacency Matrices

In addition to the morphological matrix, the incompatibility and adjacency matrices have are initially generated in a spreadsheet format and then transferred into a text-based file, where it can be read from the concept and technology exploration environment. The tasks leading to the implementation of each matrix is described in the following paragraphs.

The incompatibility matrix (M_I) is a simple way to model incompatibilities between concepts. Its general form is displayed in Equation 5.3, where n represents the number of concepts, and $\sigma_{i,j}$ has a value of 0 if concepts i and j are incompatible, and a value of 1 otherwise. It is a symmetric matrix because if concepts i and j are incompatible, then concepts j and i are incompatible as well.

$$M_I = \begin{bmatrix} \sigma_{1,1} & \cdots & \sigma_{1,n} \\ & \ddots & \\ \vdots & \sigma_{i,j} & \vdots \\ & & \ddots \\ \sigma_{n,1} & \cdots & \sigma_{n,n} \end{bmatrix} \tag{5.3}$$

The process to construct the incompatibility matrix requires the human intervention because there are different types of incompatibilities, which would be too complex to model by logical operators. For instance, some incompatibilities are related to physical constraints (e.g., liquid rocket engines cannot use solid propellant), some others are related to constraints of the modeling and simulation environment

INCOMPATIBILITY MATRIX			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
			OVERALL									BOOSTERS								
			Number_stages		Boosters		Launch Cycle		Number_stages			Engine_type		Propellant_type			Burn_time			
			1	2	no	yes	vertical	horizontal	stacked	piggy_back	internal	liquid	solid	lox_lh2	lox_rp1	ap_htpb	100	125	150	200
1		Number_stages	1		0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1
2			2		0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3		Boosters	no		1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0
4			yes		1	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1
5	Overall Vehicle	Launch_cycle	vertical		1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1
6			horizontal		0	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0
7		Stage_formation	stacked		1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1
8			piggy_back		1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1
9			internal		1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1
10		Engine_type	liquid		1	1	0	1	1	0	1	1	1	0	0	1	1	1	1	1
11			solid		1	1	0	1	1	0	1	1	1	0	0	0	1	1	1	1
12		Propellant_type	lox_lh2		1	1	0	1	1	0	1	1	1	1	0	0	0	1	1	1
13			lox_rp1		1	1	0	1	1	0	1	1	1	1	0	0	0	1	1	1
14	Boosters		ap_htpb		1	1	0	1	1	0	1	1	1	0	1	0	0	1	1	1
15		Burn_time	100		1	1	0	1	1	0	1	1	1	1	1	1	0	0	0	0
16			125		1	1	0	1	1	0	1	1	1	1	1	1	0	0	0	0
17			150		1	1	0	1	1	0	1	1	1	1	1	1	0	0	0	0
18			200		1	1	0	1	1	0	1	1	1	1	1	1	0	0	0	0

Figure 5.2: Incompatibility matrix example

(.e.g, an airbreathing launch vehicle cannot be modeled with boosters), whereas some other incompatibilities are related to common sense (e.g., one cannot select simultaneously 2 and 3 boosters on the vehicle). The thinking process of the designer can more easily scan through these incompatibilities, and build the incompatibility matrix manually.

To help this process, an algorithm in Visual Basic is created, which requires that only the upper part of the incompatibility matrix be filled with incompatibilities. The algorithm then automatically creates the incompatibility by filling the empty cells by ones, as well as the lower part of the matrix. Figure 5.2 shows an example of incompatibility matrix for a launch vehicle. Notice that the diagonal of the incompatibility, as well as the concepts grouped in a same morphological field are zero. Indeed, a concept cannot be selected twice, and only one concept can be selected per morphological field in the morphological matrix. For example, an architecture cannot have one *and* two stages.

The incompatibility matrix of the morphological matrix in Table 5.1 can then be represented by Equation 5.4. Notice the incompatibilities between the concepts from the same morphological field. Moreover, it is assumed that RBCC propulsion

systems cannot be combined with boosters, hence showing the other 0 values in the incompatibility matrix.

$$M_I = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{5.4}$$

Similar to the incompatibility matrix, the adjacency matrix is created for modeling the combinations between alternatives in a design problem. Unlike the incompatibility matrix, the adjacency matrix (M_A) represents the possible arcs that can be used to go from one node to an other. More specifically, these arcs represent a combination between two concepts, which are themselves represented by two specific nodes. It is thus represented by Equation 5.5, where n is the number of concepts, and $\alpha_{i,j}$ has a value of 1 if a combination is compatible between two concepts i and j, or a value of 0 otherwise. The adjacency matrix is similar to the incompatibility matrix since it is a symmetrical, square matrix of a size equal to the number of concepts in the morphological matrix. Its only difference is that it does not require to track the decisions made previously. That is, a value of $\sigma_{i,j} = 0$ in the incompatibility matrix means that concepts i and j cannot be on the same architecture at any time. However, a value of $\alpha_{i,j} = 1$ means that a move from concept i to concept j is possible. It does not eliminate the possibility of finding concepts i and j in the same solution. For example, if concept 1 is incompatible with concept 2, the incompatibility matrix indicates that two concepts cannot be grouped, at any time. In the case of the graph,

if there is no link between concepts 1 and 2 but concept 1 is selected, concept 2 can still be selected after concept 3 if the moves from 1 to 3 and 3 to 2 are possible. To put it simply, the adjacency matrix does not take into account the previous moves but the incompatibility matrix does, yet they are both needed in the concept exploration phase. This will be shown in Chapter 7. Before, however, the concept string must be passed to the modeling and simulation environment, which is explained it the next section.

$$M_A = \begin{bmatrix} \alpha_{1,1} & \cdots & \alpha_{1,n} \\ & \ddots & \\ \vdots & \alpha_{i,j} & \vdots \\ & & \ddots \\ \alpha_{n,1} & \cdots & \alpha_{n,n} \end{bmatrix} \tag{5.5}$$

5.3.3 Task 3: Concept String to Morphological Data Structure Procedure

As a reminder, the overall goal of this section is to show how the alternatives in the morphological matrix are mapped to the modeling and simulation environment, that is, Step 4 of the new design process. The third and last task of Step 4 is to convert the architecture, expressed in a binary concept string (see Equation 5.1), into an appropriated settings of input variables that are required for the modeling and simulation environment. To perform this task, a transformation is executed, which interprets the concept string and generates a structured set of input variables.

The first part of this transformation is to determine the concepts and technologies selected from concept string. This is done by scanning each of the bit in the concept string that contains a value of 1 and to associate the concept to a digit, as shown in Table 5.2. A vehicle architecture then results of this operation where an alternative is assigned for each morphological field in the morphological matrix. For example, the concept string of Equation 5.2 would correspond to the following data structure

representing the architecture.

```
Morph.vehicle_type = wing_body;
Morph.engine_type = liquid_rocket;
Morph.number_of_boosters = 2;
```

This approach has two major advantages. First, it organizes the concepts selected in morphological fields, and second, it helps the transition of the architecture to the modeling and simulation environment, as discussed in the next paragraph.

The second part of this task consists of adapting the morphological data structure to the appropriate set of input variables required to run the modeling and simulation environment. This task is done by mapping each design variable to a concept. For example, the specific impulse of a liquid engine is function of the propellant type. Thus, Algorithm 1 could be used to map the specific impulse as a function of the propellant type. Similarly, Algorithm 2 could be used to determine the final velocity of the architecture, considering that one or two stage vehicle stages have different final velocities. Repeating this approach for all the design variables required for the modeling and simulation environment to build an entire input file for each architecture generated.

Input: fuel
Output: Engine specific impulse in vacuum
if *engine type* = *liquid* **then**
  switch *fuel type* **do**
    case *LOX-LH2*
      Prop.Isp = 455 s;
    case *LOX-RP1*
      Prop.Isp = 353 s;
    case *htpb*
      Prop.Isp = 286 s;
  end
end

Algorithm 1: Concept mapping to the modeling and simulation

Input: Morphological matrix
Output: V

if *Number of stages = 1* **then**
    Stage(1).Final velocity = 7784 m/s ;
else
    Stage (1).Final velocity = Morph.Second stage.Final velocity ;
    Stage (2).Final velocity = 7784 m/s ;
end

Algorithm 2: Concept mapping to the modeling and simulation

Table 5.3: Description of the modeling structured array

Structure Field	Description
Mission	Mission parameters
Config	Overall vehicle configuration
Prop	Propulsion parameters and configuration
Geom	Geometry parameters
Mass	Mass estimating parameters
Cost	Cost estimating parameters
K	Technology factors

To facilitate the handling of a large number of input variables, they are grouped in a structured array and the mapping approach mentioned above can now be applied by variable groups, which simplifies the process. A total of seven input variable sets are created, as depicted in Table 5.3 with their respective description.

As a result of the three tasks just described, any architecture defined by a concept string can be evaluated quantitatively. This is an essential element in achieving this objective which is to enable the quantitative design space exploration or aerospace alternatives. The next step consists of enabling the automatic generation of compatible architectures without requiring human intervention. This issue is assessed by a solution construction technique presented in the next section.

5.4 Step 5: Generate Compatible Architectures

This section shows how compatible solutions can be generated computationally from the morphological matrix. To achieve this goal, the morphological matrix must be

Table 5.4: Notional Morphological Matrix

	Concepts		
Vehicle type	Wing body	Lifting body	
Engine Type	RBCC	Liquid rocket engine	Solid rocket engine
Number of boosters	0	2	3

transformed into a graph. This is done by replacing the concept in the morphological matrix by nodes and by replacing the possible combinations of concepts by edges. Taking the notional morphological matrix presented in Table 5.4 leads to the morphological graph depicted in Figure 5.1. The graph shows eight different concepts and the possible combinations between concepts. It is interesting to note that no link exist between two concepts of the same morphological field. For example, there is no line between RBCC and liquid rocket because either one or the other must be selected. This graph is the first step in the development of the solution construction approach for any launch vehicle architecture, as detailed in the next paragraphs.

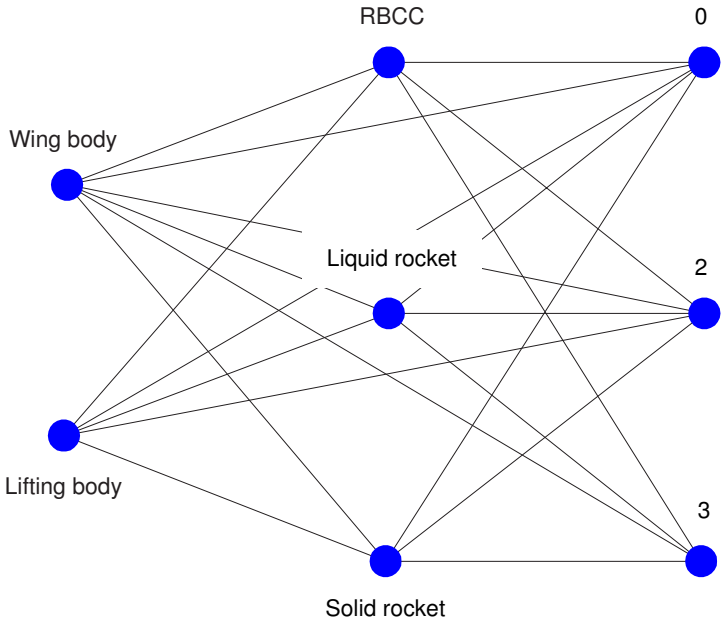


Figure 5.3: Graph of the morphological matrix presented in Table 5.4

As observed in Figure 5.3, searching over a graph could require human interaction

to differentiate between the nodes and vertices. Despite this intuition, there exist two graph searching techniques that enable the easy computational exploration of a graph. The first technique is a branch and bound method [61] where certain sub-regions are selected separately (branch), and these sub-regions are then bounded together to create the solution. The second technique constructs a solution by sequentially going from node to node in a stepwise manner. This second technique prevails over a branch and bound technique because of the hierarchical nature of the architecture selection problem. In other words, the concept selection decisions are sequential and cannot be performed in parallel without the sharing of information between the various branches. This approach is detailed in the next paragraph.

The solution construction requires the definition of the adjacency matrix, which is a way to display the links between nodes of a graph. The adjacency matrix, as explained earlier, is a square matrix that contains a value of 1 if two nodes are linked and 0 otherwise. This transformation of a graph into a binary matrix is the key for a computer program to generate paths automatically. The morphological matrix of Table 5.4 is represented by the adjacency matrix of Table 5.5, where each row and column are associated to the concepts in the table note. These concepts are also the same ones used throughout this chapter for consistency.

Once the adjacency matrix is built, an algorithm can generate a solution from the morphological graph. This is done by moving from a node in a morphological field to another, and by following the vertices in the morphological graph. A launch vehicle architecture is then constructed when at least one concept from each morphological field is selected. In terms of programming, the path is constructed by starting from one entry in the adjacency matrix and moving to another entry on the same row of the matrix with a value of 1. For example, if the first entry selected is the row A1, then the available moves in row A1 are the columns B1 to C3. In more technical terms, this means that the wing body vehicle type (row 1) can be linked to any concept in

Table 5.5: Adjacency matrix of the morphological matrix in Table 5.4*

	A1	A2	B1	B2	B3	C1	C2	C3
A1	0	0	1	1	1	1	1	1
A2	0	0	1	1	1	1	1	1
B1	1	1	0	0	0	1	1	1
B2	1	1	0	0	0	1	1	1
B3	1	1	0	0	0	1	1	1
C1	1	1	1	1	1	0	0	0
C2	1	1	1	1	1	0	0	0
C3	1	1	1	1	1	0	0	0

* A1 = wing body, A2 = lifting body, B1 = RBCC, B2 = liquid rocket engine, B3 = solid rocket engine, C1 = no booster, C2 = 2 boosters, and C3 = 3 boosters

the second or third morphological field of the morphological matrix (Table 5.4). This process is repeated until a complete vehicle is generated. This concludes the final step in the solution construction technique, that is, Step 5 of the new architecture selection process presented in Chapter 3. The final step consists of evaluating the architecture performance, which is described partially in the next section, and more thoroughly in Chapter 6.

5.5 Step 6: Sizing the Architecture in the Modeling and Simulation Environment

5.5.1 Concept Modeling

The goal of Step 6 is to evaluate quantitatively the different vehicle architectures. This step is vehicle-specific since every vehicle type is modeled differently. A commercial airplane, for example, cannot be modeled the same way as a launch vehicle. Yet, the modeling and simulation must be as general as possible to enable a large variety of system architectures. Thus, a choice must be made in the type of vehicle to evaluate.

For this thesis, launch vehicles are selected as the system of choice. In fact, launch vehicles have a large variety of concepts and thus are a good vehicle to apply the new

method. Consequently, a modeling and simulation environment was developed to ensure that a single code would enable the evaluation of the large combination of launch vehicle architectures. This code is named Rapid Access-to-Space Analysis Code-2 (RASAC-2), and its formulation is described in Chapter 6.

5.5.2 Technology Modeling

The traditional approach in conceptual design consists of selecting the technologies after the vehicle architecture is defined. The approach requires design engineers to estimate the technology impact on reduction factors (K-factors) in the modeling and simulation environment. The technology space is then explored by evaluating certain combination of technologies under economic constraints. [56] The technology space is usually explored using stochastic optimization algorithms [77], and the technology reduction factors (k-factors) are usually modeled using response surfaces to accelerate the evaluation of technology combinations. [57] The approach used in this thesis shows some differences and similarities with the traditional technology selection approach, as they are developed in the following paragraphs.

The differences with the traditional approach is in the technology representation. Indeed, the new design process groups technologies in the same morphological matrix as the concepts. There are three main advantages of this approach. The first one is that it helps to explore concepts and technologies simultaneously, which is one of the objectives of this thesis. The second one is that it helps to integrate the technologies in the graph and incompatibility matrices. This integration enables the exploration of compatible combinations of technologies and concepts using graph-based combinatorial optimization algorithms. Finally, the third advantage is that the grouping of incompatible technologies in morphological fields reduces the size of the design space. These advantages have basically no computational cost associated with them because the design process sees the technologies identically as the concepts. In

Table 5.6: Notional Morphological Matrix

Morp. field	**Concepts**		
Vehicle type	Wing body	Lifting body	
Engine Type	RBCC	Liquid rocket engine	Solid rocket engine
Number of boosters	0	2	3
Propulsion	T1	T2	
Structure	T3	T4	T5

other words, they both are discrete states in the design space. This approach helps the optimization algorithm to find good technology and concept combinations that would be eliminated using the traditional sequential approach discussed earlier. Table 5.6 shows the morphological matrix, previously presented, with five notional technologies (T1 to T5), and two new morphological fields for the technologies (Propulsion and Structure).

Once the technologies are organized in the morphological matrix, the next step consists of creating the technology impact matrix. The technology impact matrix[56] has the same form as Equation 5.3, where $\kappa_{i,j}$ represents the reduction factor i from technology j, m is the number of reduction factors, and n the number of technologies.

$$TIM = \begin{bmatrix} \kappa_{1,1} & \cdots & \kappa_{1,n} \\ & \ddots & \\ \vdots & \kappa_{i,j} & \vdots \\ & & \ddots \\ \kappa_{m,1} & \cdots & \kappa_{m,n} \end{bmatrix} \tag{5.6}$$

The technology impact matrix is used to transform the technologies "on" and "off" settings into technology reduction factors (K-factors), which are then used to assess their performance. This computation procedure is illustrated by Equation 5.7,

$$[K]^{mx1} = [TIM]^{mxn} \cdot [S_T]^{nx1} \tag{5.7}$$

where K is the vector of reduction factors, and S_T is the technology string, n is the

number of technologies. The technology string (S_T) is identical to the concept string where a value of 0 corresponds to a technology "off", and of 1 to a technology "on". This then closes the loop for the generation of compatible architectures where now the concepts and technologies can be explored simultaneously. This addresses the third research question, which wondered of the best way of to explore the concepts and technologies simultaneously. The next section describes the optimization algorithm approach to help the design space exploration.

5.6 Design Space Exploration: ACO

For this thesis, it is hypothesized that Ant Colony Optimization (ACO) is the best choice for optimizing the morphological graph. Its underlying theory was presented in Section 5.6 in a broader perspective. The formulation of the algorithm for the exploration of aerospace architectures is presented in the following sections.

5.6.1 Overall Algorithm

The overall work-flow of ACO is illustrated in Figure 5.4. The figure shows the three major steps each ant follows during the optimization, that is, construct a solution, evaluate the performance, and deposit pheromone on trails used. Moreover, a fourth operation, pheromone evaporation, is implemented to reproduce the real evaporation phenomenon occurring in nature.

This diagram in Figure 5.4 can be translated into Algorithm 3 to illustrate the programming structure used for the technology and concept selection problem. Two loops are shown: one performed for each ant and one for each iteration. The inner loop constructs the architectures, evaluates the solution, and updates the pheromone matrix based on the performance of each solution. The outer loop represents the global iteration of the algorithm, recording the best solutions, evaporating the pheromone level on the paths, and checking for convergence.

The algorithm stops when one of the conditions is met: 1) maximum number of

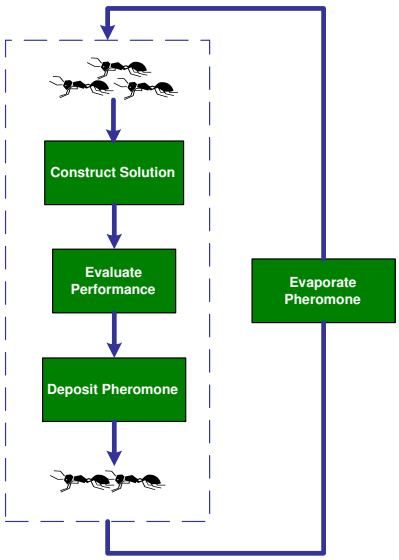


Figure 5.4: Overall ACO flow diagram

Input:
ρ = Evaporation rate ;
τ_0 = Initial pheromone level ;
A = Adjacency matrix ;
while *Termination* **do**
  for k = *1 to Number ants* **do**
    Ant Colony Paths (k) = Construct Solution(Number concepts) ;
    Performance (k) = Evaluate Performance(Ant Colony Paths (k)) ;
    A = Update Pheromone Matrix(Ant Colony Paths (k),Performance (k)) ;
  end
  Iteration best ant,Overall best ant = Record best solutions(Performance) ;
  A = Evaporate Pheromone() ;
  Termination = Check Convergence() ;
end

Algorithm 3: Ant Colony Optimization

Table 5.7: ACO termination criteria

Condition	Basline Value
Maximum number of iterations without improvement	6
Maximum number of iterations	20
Maximum time	5 hours

iterations, 2) maximum number of iterations without improvement, and 3) maximum time. Most of the time, the algorithm stops when the maximum number of iterations without improvement of the objective function is reached. Table 5.7 shows the three conditions with their respective settings for the concept and technology selection problems. The next sections details each of the steps necessary to achieve ACO.

5.6.2 Solution Construction and Performance Evaluation

The first step of Ant Colony Optimization consists of constructing solutions from the design space. Section 5.4 showed the steps to construct a solution from an adjacency matrix. The process of constructing a solution in the case of the ACO is similar with the main difference that the pheromone matrix is used to determine the moves on the graph. The use of the pheromone matrix thus implies that as the optimization progresses, certain paths will be more likely to be followed than others, as explained in the next paragraph.

In the case of a general graph, the move from node i is performed by first identifying the edges attached to the node. These edges, with the exception of the one just used to get to node i, represent the set of possible moves.

It was shown that for a regular graph, the probability of using the edge $i - j$ is equal to Equation 5.8, where p_{ij} is the probability of moving from node i to node j, N_i is an edge connected to node i, and j is an arbitrary node. The equation thus states that the probability from moving from one node to the next is uniform, if and only if, there is a link connecting the two nodes.

$$p_{ij} = \begin{cases} \frac{1}{\sum_{l \in N_i}} if j \in N_i \\ 0 \qquad\qquad if j \notin N_i; \end{cases} \tag{5.8}$$

When using a pheromone matrix, rather than an adjacency matrix, the procedure is slightly different. This difference arises because the probability of moving from a node to another is not uniform but proportional to the quantity of pheromone on an edge. Thus, in ACO, the probability of moving from node i to node j is represented by Equation 5.9, which was already introduced in Chapter 4.

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{l \in N_i^k} \tau_{il}^\alpha \cdot \eta_{il}^\beta} & if j \in N_i^k \\ 0 & if j \notin N_i^k; \end{cases} \tag{5.9}$$

For Equation 5.9, p_{ij}^k is the probability for an ant k located at the node i to go to node j, and τ_{ij} is the pheromone concentration of arc (i, j). In addition, N_i^k represents the neighborhood (i.e., feasible moves) of an ant k at node i, l the feasible nodes for an ant k, and α the relative importance of pheromone level over the heuristic information (β) contained in η_{ij}.

Since no heuristic information exist between the two nodes of a morphological graph a second transformation is done. Setting $\beta = 0$ and $\alpha = 1$, Equation 5.9 becomes equal to Equation 5.10, shown below.

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}}{\sum_{l \in N_i^k} \tau_{il}} & if j \in N_i^k \\ 0 & if j \notin N_i^k; \end{cases} \tag{5.10}$$

Equation 5.10 states that the probability of moving to a different node is proportional to the amount of pheromone on the edge. It also takes into account incompatibilities and visited nodes by tracking N_i^k, which represents the possible moves of an ant k on node i. This equation is thus applied to each node until the morphological graph has selected a concept for each morphological field. This process is repeated

for each ant in the colony, and for each iteration performed by the ACO algorithm. This process is represented by Algorithm 4.

```
for k = 1 to Number ants do
    Path (1)^k = Find starting node();
    while i ≤ Number morph. fields do
        Path (i)^k = Find next node();
        N_i^k = Update visited nodes();
    end
end
```

Algorithm 4: Ant Colony Optimization

Once a path is selected, the solution is evaluated using RASAC-2, the modeling and simulation environment. The procedure, presented in Section 5.3 is followed to transform the solution represented in a graph format to the correct input file representing the architecture selected. This process is thus repeated for every ant until the algorithm finishes. The algorithm records all the solutions and their respective performance to keep track of the solutions evaluated. This is detailed in the next section.

5.6.3 Statistics Computation and Pheromone Update

During the optimization of the algorithm, the overall-best alternative, the iteration-best alternative, and the pheromone matrix are updated at the end of each iteration. First, the overall best ant and the iteration best ant are recorded for each iteration, based on their respective value of the objective function. A pheromone quantity is then deposited on the graph by means of updating the pheromone matrix. This operation is displayed by Algorithm 5. The algorithm states that after each tour, the amount of pheromone deposited by an ant k is on all the edges i-j visited is proportional to the goodness of the solution.

The final step then consists of performing the pheromone evaporation on the pheromone matrix. This step is simply performed by reducing the pheromone level in the pheromone matrix by a ratio ρ. This is illustrated by Algorithm 6.

Input:
F_{Ref} = Referenced value of the objective function ;
Ph = Pheromone matrix
i = Concept string (1) ;
for *k = 1 to Number ants* **do**
for *l = 2 to Number of concepts* **do**
j = Concept string (l) ;
$\Delta\tau = F_{Ref}$/ F (k) ;
Ph (i,j) = Ph (i,j) + $\Delta\tau$;
Ph (j,i) = Ph (i,j) ;
i = j;
end
end

Algorithm 5: ACO pheromone deposition

Input:
ρ = Pheromone evaporation rate ;
Ph = Pheromone matrix ;
for *i = 1 to Number of concepts* **do**
for *j = i to Number of concepts* **do**
Ph (i,j) = Ph (i,j) · (1-ρ) ;
Ph (j,i) = Ph (i,j) ;
end
end

Algorithm 6: ACO pheromone evaporation

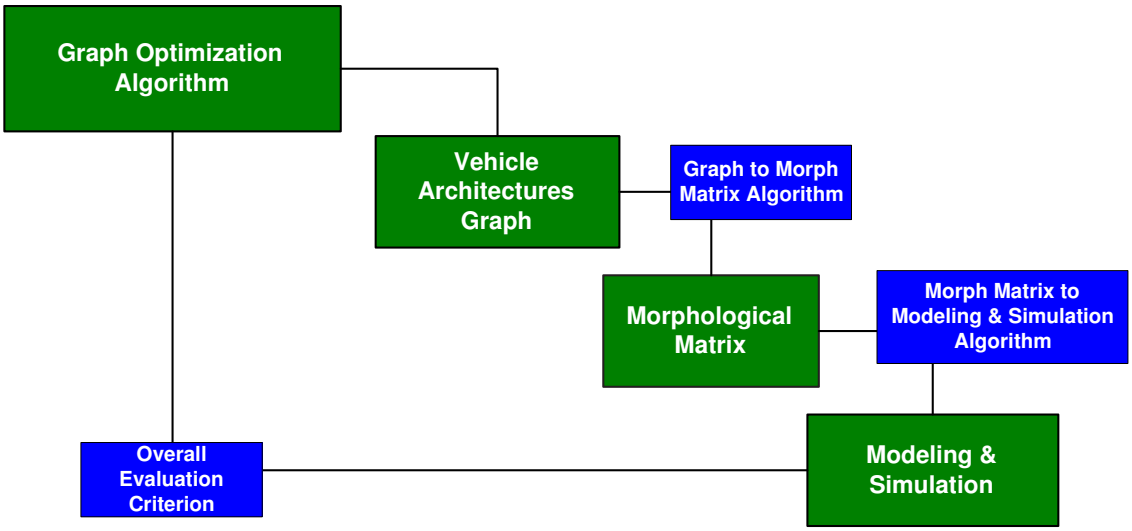


Figure 5.5: Overall concept and technology exploration process

The same process as detailed for ACO is applied to run TS. The only exception is that TS does not update a pheromone matrix. It only uses the adjacency matrix, which means that the selection of the moves is uniformly distributed on all the edges $l \in N_i$ is a set of possible moves.

As a summary, Figure 5.5 shows the overall work-flow required to perform a graph-based optimization. It shows that the optimization algorithm generates solutions on a graph which are then interpreted in the morphological matrix to generate an architecture array. This information is sent to the modeling and simulation environment, which sends the information back to the optimization algorithms via an overall evaluation criterion. This loop is repeated for each solution generated by the algorithm.

The graph optimization algorithm is the last element in the implementation of the new design process presented in Chapter 3. The next chapter thus presents the formulation of the launch vehicle sizing code, which is an enabler to the validation and the verification of the new design process presented in this thesis.

CHAPTER VI

FORMULATION: LAUNCH VEHICLE MODELING AND SIMULATION

This chapter presents the formulation of the modeling and simulation environment for launch vehicle architectures. The modeling and simulation environment is required to perform the sixth step of the concept and technology down-selection methodology, detailed in Chapter 3. It is developed as a means to demonstrate this methodology, and it uses launch vehicle data and models that are publicly available.

The philosophy behind the development of this launch vehicle modeling environment resides in the capability of modeling launch vehicle concepts and technologies at the component level, and its adaptability with Ant Colony Optimization (ACO). The algorithm then uses the modeling environment to compute the goodness of the constructed solutions during the optimization process. The modeling environment has the capability of computing the performance of all the concepts and technologies under study, which is different than the traditional approach where the concepts are individually optimized. This approach also enables to simultaneously down-select the concept and technologies by bringing a higher level of fidelity into the architecture study process.

6.1 Program Structure

A design variable structure composed of four levels is used to handle the launch vehicle architectures, as shown in Table 6.1. The first level represents the number of stages. One and two stages are modeled. The second level represents the two vehicle types that can be used on each stage, that is, 1) core stage, 2) boosters.

The propulsive stages represent the core of the launch vehicle. They have their own engines and tanks. The boosters are systems that are added to the propulsive stages usually to provide additional thrust. This approach thus enables the definition of a large number of architectures. For example, the CEV launch vehicle developed recently by NASA is composed of two core stages, whereas the CaLV is composed of two core stages and 1 set of boosters attached to the core first stage. Finally, the Level 3 and 4 represent the actual discipline and input variables for each discipline, respectively. Thus, for each of the Level 2 variables is a set of variables (Level 4) organized into disciplines (Level 3). This nomenclature helps model a larger variety of launch vehicle architectures by organizing the type of vehicle and synthesizing the overall launch vehicle, which can then be sized by the same modeling and simulation environment. The next section describes its overall structure.

Table 6.1: Design variable structure in the architecture modeling

Level	Variables
1	Stage 1, Stage 2
2	Boosters, Core
3	Mission, Configuration, Propulsion, Weight, Geometry, Cost, Technologies
4	Final velocity, payload weight, ...

To enable rapid analysis of launch vehicles, a sizing and synthesis code is developed. This code, named Rapid Access-to-Space Analysis Code 3 (RASAC-3), is the third generation of RASAC's sizing and synthesis environment.

Figure 6.1a shows the design structure matrix of RASAC-2. It is composed of six disciplinary models: 1) Aerodynamics, 2) trajectory simulation, 3) geometry, 4) propulsion, 5) mass, and 6) cost models. These models are used to compute the performance of each launch vehicle architecture generated.

The remaining of the chapter is organized according to the aerospace disciplines enumerated below. Each discipline, except for the aerodynamics, was specifically

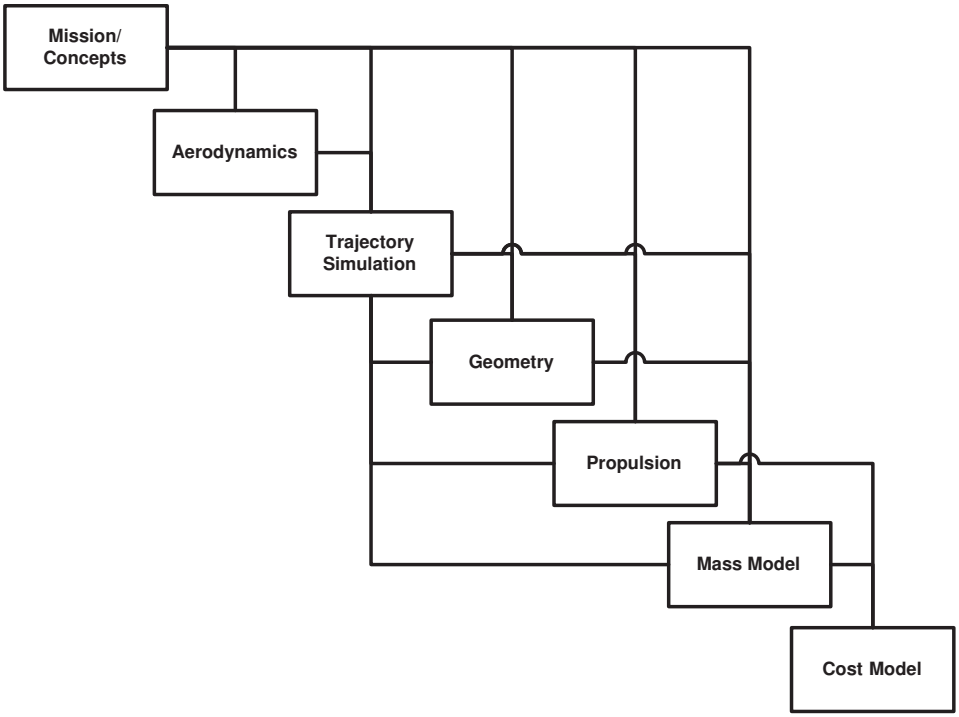


Figure 6.1: Rapid Access to Space Analysis Code 3 (RASAC-3) design structure matrix

programmed for this thesis, using existing models found in the literature.

1. Aerodynamics modeling - (Section 6.2)
2. Trajectory simulation - (Section 6.3)
3. Geometry modeling - (Section 6.4)
4. Propulsion modeling - (Section 6.5)
5. Mass modeling - (Section 6.6)
6. Cost modeling - (Section 6.7)
7. Sizing and Synthesis - (Section 6.8)

6.2 Aerodynamics Modeling

The aerodynamics is modeled by using lift and drag coefficient tables generated from the Aerodynamic Preliminary Analysis System (APAS). This code uses slender body theory as well as the source and vortex panel methods for subsonic and supersonic flows. It also computes the aerodynamics of hypersonic flows by the use of impact/shadow flow theories (e.g., Newtonian theory).[25] APAS computes relatively quickly the aerodynamics characteristics of a wide variety of vehicle shapes. The vehicle baseline geometry used for the aerodynamics is presented in Figure 6.2. A different set of aerodynamic tables is generated for the configuration with and without strap-on boosters.

Figure 6.2: Cargo Launch Vehicle geometry used for generating the aerodynamic coefficients.

6.3 Trajectory Simulation

This section describes the trajectory simulation model programmed to compute the trajectory of launch vehicles. The program was written in the MATLAB environment. For rocket vehicles, the state-of-the-art computational approach consists of performing a Newtonian trajectory optimization with the pitch rate as the independent variable. A derivation of this approach for a two-dimensional cartesian frame of reference with three degree-of-freedom trajectories is described in this section. The model assumes a flat Earth and no coriolis pseudo-force. The problem statement can be summarized as minimizing the mass ratio (m_0/m_f), where m_0 and m_f represent the initial and final masses, respectively. The problem is subject to maximum acceleration, final altitude, and final velocity constraints, as illustrated in Table 6.2,

Table 6.2: Optimization problem using the two-degree-of-freedom trajectory calculation

Minimize:	Mass ratio $= m_0/m_f$
Subject to:	$V_f - V_{target} \geq 0$
	$\gamma_{target} - \Delta\gamma \leq \gamma_f \leq \gamma_{target} + \Delta\gamma$
	$y_{target} - \Delta y \leq y_f \leq y_{target} + \Delta y$
	$n \leq n_{max}$

For Table 6.2, V represents the velocity, γ the flight path angle, x and y the range and altitude respectively, n the load factor, and subscripts $target$ and f the orbital and final conditions, respectively.

With a free body diagram similar to Figure 6.3, the equations of motion in a stationary cartesian frame of reference for this problem are described in Equation 6.1,

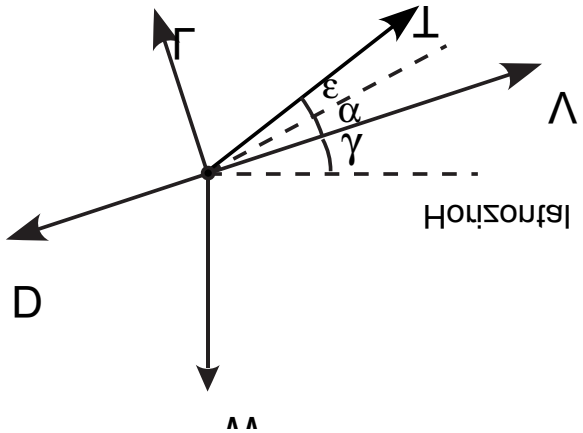


Figure 6.3: Force diagram for the trajectory simulation model

$$\begin{aligned}
\dot{x} &= u \\
\dot{y} &= v \\
\ddot{x} &= \left(-L\sin\gamma - D\cos\gamma + T\cos\tau\right)\tfrac{1}{m} \\
\ddot{y} &= \left(L\cos\gamma - D\sin\gamma - mg_0 + T\sin\tau - \tfrac{mV^2}{r}\right)\tfrac{1}{m} \\
\dot{m} &= \tfrac{-T}{g_0 Isp}
\end{aligned} \tag{6.1}$$

where u and v represents the velocity in x and y, respectively, L the lift, D the drag, T the thrust, τ the angle between the thrust vector and the velocity vector, Isp the specific impulse, and m the instantaneous mass of the vehicle.

Given that $L = f(u, v, y)$, $D = f(u, v, y)$, $T = f(y)$, $g = f(y)$ and $m = f(t)$ Equation 6.1 becomes a system of five equations and six unknowns (x, y, u, v, t, and Θ). To solve this system, a sixth equation must be introduced. The thrust vector control is thus the sixth equation for the system as described by Eq.(6.2).

$$\Theta = \Theta(t) \tag{6.2}$$

It can be shown that the optimum trajectory for a drag-free powered flight requires a linear variation of the thrust vector angle (θ) over time [97]. Similarly, a general thrust steering control is performed with segmented linear variations over time of

the thrust vector angle as shown in Figure 6.4. The angle and time values of the thrust vector angle are then optimized in order to minimize the mass ratio with final altitude, velocity, and flight path angle constraints. An explicit 4^{th} order Runge-Kutta ordinary differential equation solver can be used, for example, to solve the equations of motions presented in Equation 6.1. This description of the thrust vector angle as function of time is then used to solve and optimize in a step-wise manner the system of ordinary differential equations shown in Equation 6.1.

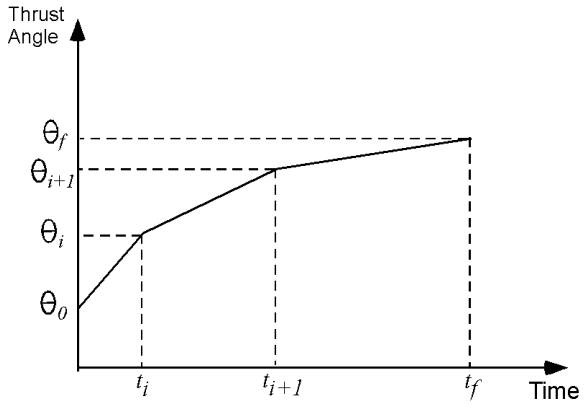


Figure 6.4: Linear thrust steering example

6.3.1 Verification

The trajectory optimization model is verified against simulations ran in POST[75], as well as the ESAS trajectory simulation for the Cargo Launch Vehicle (CaLV). The mission considered for this verification exercise consists of launching the vehicle from Cape Canaveral and attaining a 475,872 ft orbit at 1.0522 deg. The remaining of the mission requirements are detailed in Table 6.3

The CaLV consists of two LOX/LH2 stages with solid rocket boosters. The characteristics of each stage and booster are described in Table 6.4.

In addition to the propulsive characteristics, the boosters burntime is set at 125 sec, the staging velocity at 17,000 ft/s, and the fairing mass of 10,522 lbm is jettisoned after 200 s after liftoff. These parameters are summarized in Table 6.5.

Table 6.3: Mission parameters

Parameter	Value
Final altitude, ft	475,872
Final absolute velocity, ft/s	25,707
Final flight path angle, deg	1.0522
Launch pad latitude, deg	28.465 N
Trans-lunar injection propellant, lbm	236,636

Table 6.4: Input parameters

Parameter	Boosters	Core first Stage	Core second Stage
Thrust (vac), lbf	7,978,188	2,562,050	2,562,050
Isp (vac), s	266.2	452.1	451.5
Nozzle exit area, ft^2	21.62	201	68.3
Empty mass, lbm	456,665	194,563	42,528

Table 6.5: Input parameters

Parameter	Value
Boosters burntime, s	125
Staging velocity, ft/s	17,000
Fairing mass, lbm	10,522
Fairing jettison time, s	200

Table 6.6: Mass ratio comparison between RASAC3 and POST

	RASAC3	POST	Error, %
Mass ratio	16.00	16.48	-2.91

A sequential quadratic programming optimization procedure is performed on the pitch rate variation as a function of time. Matlab's *fmincon* function is used for this purpose. The trajectory is computed by integrating the 2D equations of motion with inclusion of the centrifuge pseudo-force, using Matlab's *ode45* function.

The trajectory module developed for this thesis is verified with the trajectory outputs obtained from POST using the aforementioned input parameters. The comparison between the mass ratio of both ode is displayed in Table 6.6 showing the small difference between the mass ratio computed by RASAC-3 and POST.

The control vector for both POST and RASAC-3 is the pitch rate variation as a function of time. It is compared in Figure 6.5, where one can see the similar trends between the two codes.

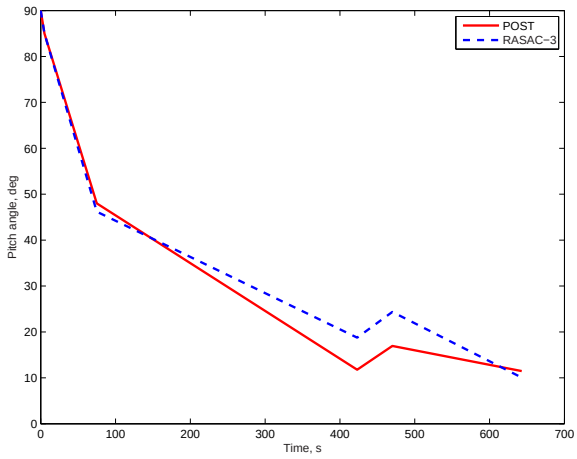
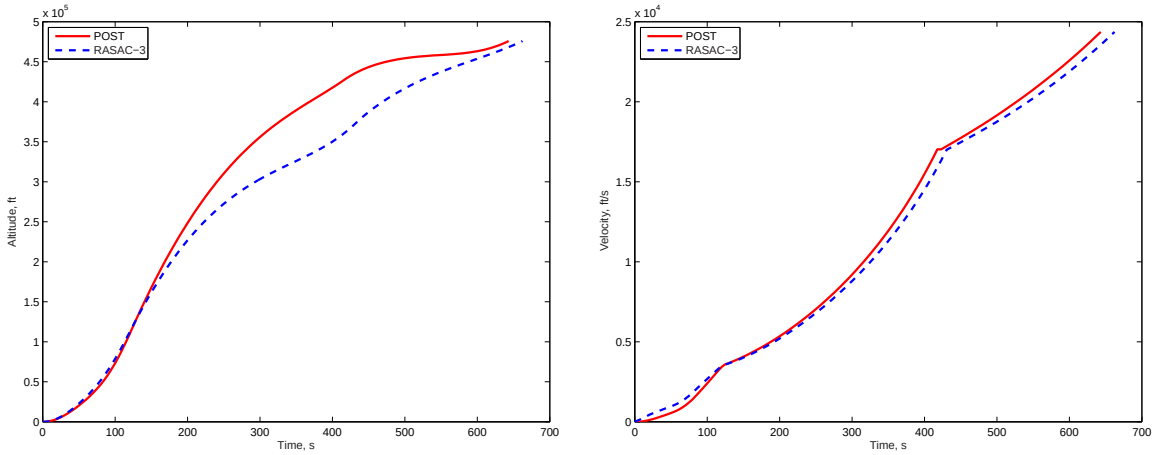


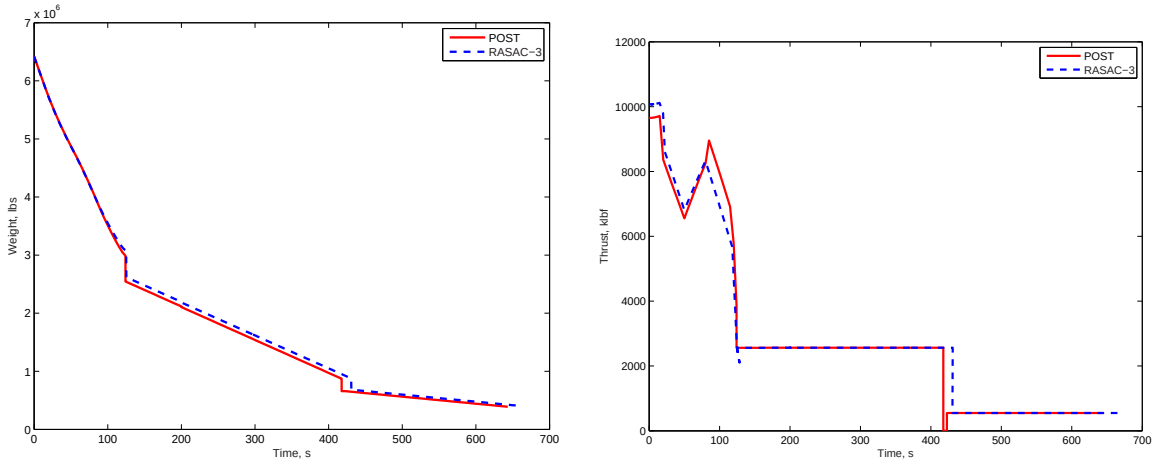
Figure 6.5: Comparison between the optimized pitch angle variation as a function of time

Additional trajectory plots are presented to compare the outputs from POST and RASAC-3. These outputs are presented in Figures 6.6 to 6.9 where the altitude, velocity, mass, thrust, drag, lift, acceleration and flight path angles are plot as function



(a) Altitude (b) Velocity

Figure 6.6: Altitude and velocity variation of the CaLV as a function of time



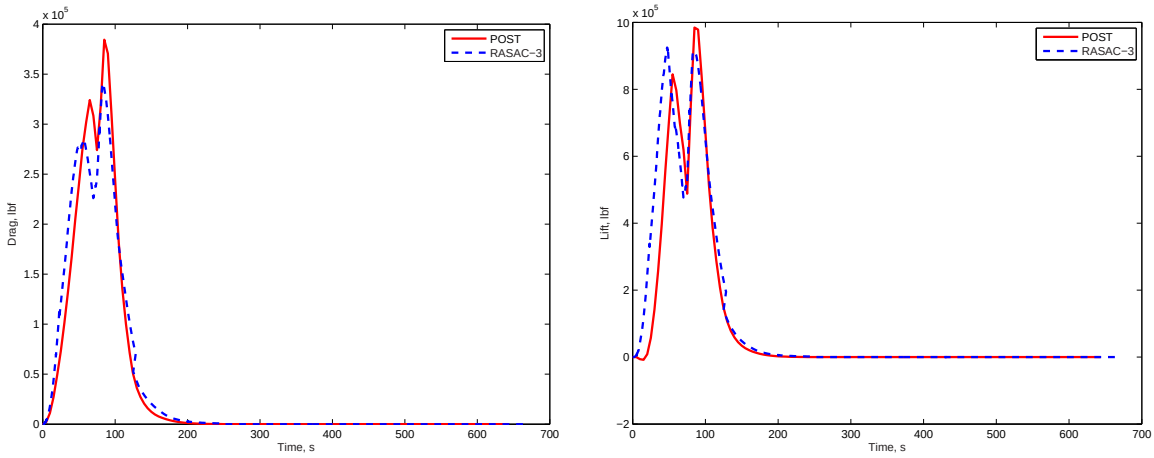
(a) Mass (b) Thrust

Figure 6.7: Mass and thrust variation of the CaLV as a function of time

of the flight time. One can observe the similarity between all the plots but the altitude variation. This is because of a different pitch rate variation, as presented in Figure 6.5.

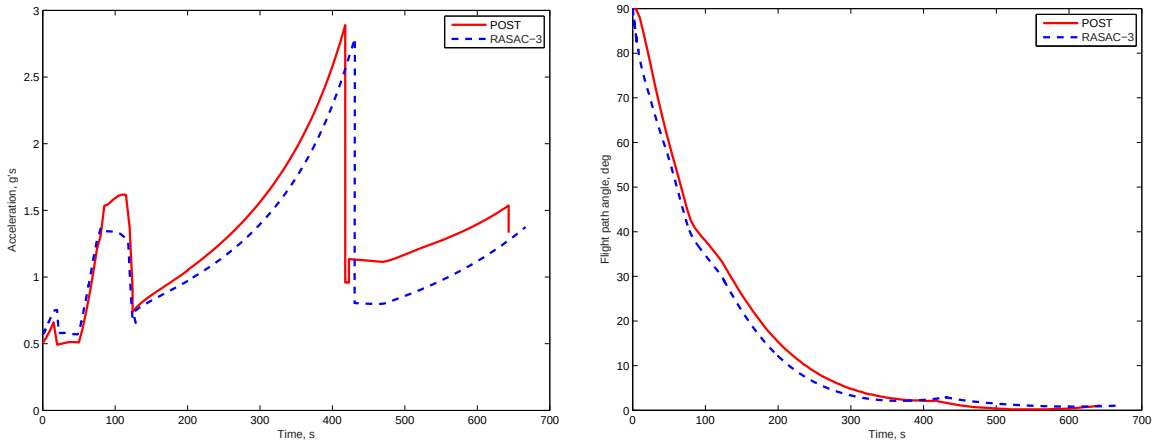
6.3.2 Response Surface Modeling

RSEs are generated to speed up the process of determining the launch vehicle performance. Four different classes of RSEs are generated, for which the ΔV required of each stage is regressed. Those four vehicle classes are selected to model appropriately the morphological matrix; they are enumerated below.



(a) Drag (b) Lift

Figure 6.8: Drag and lift variation of the CaLV as a function of time



(a) Acceleration (b) Flight path angle

Figure 6.9: Acceleration and flight path angle variation of the CaLV as a function of time

Table 6.7: Variables and ranges used for the RSE

Variable	Min	Max
Liftoff T/W	1.3	1.6
Boosters T/W (% of liftoff T/W)	0.3	0.75
Stage 2 T/W	0.65	0.8
Boosters Isp, s	266	452
Stage 1 Isp, s	266	452
Stage 2 Isp, s	266	452
Boosters staging velocity, ft/s	3,000	6,000
Stage 1 staging velocity, ft/s	8,000	15,000
Boosters structural mass ratio	5	10

1. SSTO
2. SSTO with boosters
3. TSTO
4. TSTO with boosters

Similar to Joyner and Sabatella [55], the thrust-to-weight ratio, the specific impulse, the staging velocity, and the boosters structural mass ratio are selected as the RSE input variables. Each input and its respective range used for response surface modeling is shown in Table 6.7. A central composite design of experiment, combined to random cases are used to generate the design of experiments. At first, quadratic RSEs are regressed. Then, if the error of the model fit is inappropriate, neural nets are generated. The response surface creation was performed by STARS 2.0 [59] for quadratic RSEs and BRAINN 2.0 [54] for neural nets, two Matlab programmed softwares developed at the Aerospace Systems Design Lab.

The model type, regression error, and error distribution for each stage of the four vehicle RSE classes are presented in Table 6.8. It shows that most of the vehicle stages required ΔV could be modeled by quadratic RSEs, except for SSTO, and the two stages of a TSTO launch vehicle with boosters. The table also shows the model mean square (R^2), and the Model Representation Error (MRE) mean and standard deviation. It shows that the RSE fits for each class of launch vehicle are good.

Table 6.8: RSE error

				MRE, %	
Class	**Stage #**	**Type**	**R^2**	**Mean**	**St. dev.**
SSTO	1	N.N.	0.985	-0.029	0.172
SSTO	0	Quadratic	0.998	0.017	0.560
(boosters)	2	Quadratic	0.998	-0.273	3.022
TSTO	1	Quadratic	1.000	0.024	0.213
	2	Quadratic	1.000	-0.021	0.329
TSTO	0	Quadratic	0.999	0.134	2.012
(boosters)	1	N.N.	0.988	0.368	2.955
	2	N.N.	0.982	0.421	3.365

6.4 Geometry Modeling

This section presents model programmed to represent the geometry of the launch vehicle. It differentiates between the tank and the body section of a launch vehicle stage.

6.4.1 Propellant Tank

The sizing of the propellant tank is performed by determining the tank diameter and length necessary to hold the propellant required to fly the mission. Thus, the propellant tank volume is estimated by Equation 6.3,

$$V_{tank} = 2V_{dome} + V_{linear} \tag{6.3}$$

here V_{dome} is the dome volume and V_{linear} is the volume of the linear section of the tank. Assigning a dome height of $h_{dome} = \frac{\sqrt{2}}{2}r$ where r is the tank, one can compute the dome volume using Equation 6.74, where D is the tank diameter. Moreover, one can estimate the remaining section of the tank by using Equation 6.6, where L is the tank length in its linear section.

$$V_{dome} = \frac{\pi D^2 h_{dome}}{6} \tag{6.4}$$

$$V_{linear} = \pi D \cdot L \tag{6.5}$$

6.4.2 Body

The other sections of the stage is estimated by determining the length of the forward of aft skirt, the engine compartment, the intertank, and the interstage. For this purpose, since the MERs require the surface area of the respective structure, one can use Equation 6.74 to determine its area, where S is the surfact area, D is the diameter, and L is the structure length.

$$S = \pi D * L \tag{6.6}$$

The length of the structure components are scaled geographically and in most cases, the intertank length is computed by assuming a tank clearance of 2 ft.

6.5 Propulsion Modeling

This section presents the propulsion model, programmed for computing the thrust and specific impulse during the launch vehicle flight. The model is based on solid and liquid rocket engine equations. First, the thrust, specific impulse and mass flow rate of standard rocket engines are modeled according to Equations 6.7 to 6.9, where T is the thrust, T_{vac} the vacuum thrust p_a the atmospheric pressure, A_{exit} the nozzle exit area, Isp the specific impulse, Isp_{vac} the vacuum specific impulse and g the gravitational acceleration at sea level. Equation 6.7 assumes that the engine has a Bell nozzle optimized for vacuum conditions, Equation 6.8 states that the engine Isp varies linearly with the thrust correction factor, and Equation 6.9 is another form of the definition of the specific impulse.

$$T = T_{vac} - p_a A_{exit} \tag{6.7}$$

$$Isp = Isp_{vac} T / T_{vac} \tag{6.8}$$

$$\dot{m} = -T / (gIsp) \tag{6.9}$$

The specific impulse value varies as a function of the propellant type, as well as the engine parameters. The procedure below is thus used to determine the vacuum and sea-level specific impulse for various rocket engines, where ε is the expansion ratio, c^* is the propellant characteristic velocity, γ is the ratio of specific heats, P_c is the chamber pressure, η_{engine} is the engine overall efficiency, and P_a is the atmospheric pressure.

Table 6.9: Engine sizing calculation procedure

Given:	ε, c^*, γ, P_c , η_{engine}, P_a
Find:	P_{exit}, Isp, C_T

Then, the approach described by Humble, 1995 [50] is used to determine the exit pressure of the nozzle, the engine Isp, and the thrust coefficient. First, the exit pressure is determined by solving for M_e using Equations 6.10 and 6.11.

$$\frac{P_e}{P_c} = \left\{1 + \frac{\gamma - 1}{2} M_e^2\right\}^{\gamma/(1-\gamma)} \tag{6.10}$$

$$\varepsilon = \frac{1}{M_e}\left\{\frac{2}{\gamma + 1} \cdot \left(1 + \frac{\gamma - 1}{2} M_e^2\right)\right\}^{(\gamma+1)/(2\gamma-2)} \tag{6.11}$$

Then, the thrust coefficient, defined as the thrust divided by the chamber pressure and the throat area, is calculated using Equation 6.12.[95]

$$C_T = \left[\frac{2\gamma^2}{\gamma - 1} * \left(\frac{2}{\gamma + 1}\right)^{\frac{\gamma+1}{\gamma-1}} \left(1 - \left(\frac{P_e}{P_c}\right)^{\frac{\gamma-1}{\gamma}}\right)\right]^{1/2} + \left(\frac{P_e - P_a}{P_c}\right)\varepsilon; \tag{6.12}$$

Finally, the engine specific impulse is calculated using Equation 6.13, where η_e is the engine efficiency, and g_0 is the gravitational constant as sealevel.

$$Isp = \eta_e \frac{C_T c^*}{g_0} \tag{6.13}$$

Four types of propellants are modeled in this thesis: 1) LOX/LH2, 2) LOX/RP, 3) N2O4/MMH, and 4)PBAN (solid rocket propellant). The properties for each type of propellant are described in Table 6.5.

Table 6.10: Rocket propellant properties

Propellant	γ	c*, ft/s	OFR	ρ_{ox}, lbm/ft^3	ρ_f, lbm/ft^3
LO_2/LH_2	1.26	7988	6	71.292	4.432
LO_2/RP	1.24	5921	2.56	71.292	50.567
N_2O_4/MMH	1.26	5805	2.16	90.521	54.937
PBAN	1.18	5190	-	108.344*	

* Solid fuel density

6.6 Mass Modeling

This section describes the mass model programmed for RASAC-3. It is composed of Mass Estimating Relationships (MER's), that compute the mass of each subsystem components on a launch vehicle stage. The mass model uses publicly available statistical mass estimating equations, which have a lower level of fidelity than restricted models. Future improvements of this could be done by looking for more accurate models, unavailable to the author at the time. This problem, however, does not interfere with the global objective of the thesis, which is to demonstrate the concept and technology down-selection methodology. Accuracy of the models can be improved with no consequence on the effectiveness of the proposed methodology.

Each launch vehicle stage is divided in various subsystems, for which a set of mass estimating relationships exists. The various subsystems considered for this thesis are listed in Table 6.11. The MERs used for each of these subsystems are described in the following sections. Unless othwerwise noted, the MERs described in the following sections are obtained from Rohrschneider, 2002[82].Moreover, the selected MERs are calibrated against the CaLV, as this vehicle is used for the demonstration of the concept and technology down-selection methodology.

6.6.1 Model Description

1 - Body Group

The body group MERs compute the body mass for expendable vehicles using liquid (Section 6.6.1) and solid rockets (Section 6.6.1). Each approach is described below.

Liquid Rocket Vehicles The body mass for a liquid rocket stage is divided into several subsystems, as displayed in Equation 6.14, where the variables are defined in Table 6.12.

Table 6.11: Subsystem breakdown for mass estimation

#	Mass group
1	Body
2	Main propulsion
3	Reaction Control System (RCS)
4	Orbital Maneuver System (OMS)
5	Primary power
6	Electrical Conversion and Distribution (ECD)
7	Hydraulics
8	Avionics
9	Environment control
10	Personnel equipment
11	OMS/RCS on-orbit prop.
12	Residual propellants
13	OMS and RCS reserve propellants
14	RCS propellants
15	Ascent reserve propellants
16	Inflight losses
17	Startup losses

$$\begin{aligned} M_{body} &= M_{tanks} + M_{antivortex} + M_{slash\ baffles} + M_{intertank} + \\ &\quad M_{interstage} + M_{fwd\ skirt} + M_{eng\ comp} + M_{attch} + M_{thrust} \end{aligned} \tag{6.14}$$

First, the oxidizer and fuel tank masses are estimated using Equation 6.15, where two different MERs are available for pump-fed or pressure-fed propulsion systems. The pressure-fed propulsion systems requiring heavier tanks because of the higher pressure in the tanks.

$$M_{tank} = \begin{cases} (2.44 - 0.007702\rho)\,(V_{tank})^{0.8548+0.0003189\rho} & \text{(Pump-fed)} \\ (1.3012 + 0.0099P)\,(V_{tank})^{0.8647P^{0.01645}} & \text{(Pressure-fed)} \end{cases} \tag{6.15}$$

The anti-vortex and slash baffles masses are estimated from Equations 6.16 and 6.16, respectively. The mass of the anti-vortex parts in the tank is mainly function of the propellant mass flow rate and density, whereas the mass of slash baffles is function of the propellant volume and densities and the body diameter.

Table 6.12: Description of the parameters used for computing the body weight of liquid rocket vehicles

Parameter	Description
ρ_f	Fuel density, lbm/ft^3
ρ_{ox}	Oxidizer density, lbm/ft^3
D_{body}	Body diameter, ft
$\dot{m}_f$	Fuel mass flow rate, lbm/s
$\dot{m}_{ox}$	Oxidizer mass flow rate, lbm/s
$M_{antivortex}$	antivortex mass, lbm
M_{attch}	Booster attachments mass, lbm
$M_{booster}$	Booster gross mass, lbm
$M_{eng\ comp}$	Engine compartment mass, lbm
$M_{fwd\ skirt}$	Forward skirt mass, lbm
M_{fuse}	Fuselage mass, lbm
$M_{interstage}$	Interstage mass, lbm
$M_{intertank}$	Intertank mass, lbm
$M_{slash\ baffles}$	Slash baffles mass, lbm
M_{tank}	Tank mass, lbm
M_{thrust}	Thrust structure mass, lbm
$N_{booster}$	Number of boosters
P	Tank pressure, psf
Q_{max}	Maximum dynamic pressure, psf
S_{body}	Body wetted area, ft^3
$S_{eng\ comp}$	Engine compartment wetted area, ft^3
$S_{fwd\ skirt}$	Forward skirt wetted area, ft^3
$S_{intertank}$	Intertank wetted area, ft^3
$S_{interstage}$	Interstage wetted area, ft^3
V_{tank}	Tank volume, ft^3

$$M_{antivortex} = \dot{m}_f/\rho_f\,(0.64 + 0.0184\rho_f) + \dot{m}_o x/\rho_{ox}\,(0.64 + 0.0184 * \rho_{ox}) \quad (6.16)$$

$$M_{slash\ baffles} = 6.77e-7 D_{body}/1.01 \cdot (V_{tank_f}\rho_f^2 + V_{tank_{ox}}\rho_{ox}^2); \quad (6.17)$$

The inter-tank MER is described by Equation 6.18, showing that the inter-tank surface area and body diameter are the two parameters used to compute the inter-tank mass. Moreover, the inter-tank MER varies as a function of the stage number and the number of stages, which explains the three variations of the inter-tank MERs.

$$M_{intertank} = \begin{cases} S_{intertank}\ 26.36\ D_{body}^{-0.5169} & \text{(SSTO)} \\ S_{intertank}\ 27.04\ D_{body}^{-0.5169} & \text{(TSTO-Stage 1)} \\ S_{intertank}\ 21.47\ D_{body}^{-0.6025} & \text{(TSTO-Stage 2)} \end{cases} \tag{6.18}$$

Similar to the inter-tank, the interstage, forward skirt, and engine compartment MERs are function of their respective surface area and the body diameter. The designer also chooses different MERs depending on the number of stages and the stage number of vehicle. This is shown by the formulation of Equations 6.19 to 6.22

$$M_{interstage} = \begin{cases} S_{interstage}\ 17.92\ D_{body}^{-0.4856} & \text{(SSTO)} \\ S_{interstage}\ 18.57\ D_{body}^{-0.4856} & \text{(TSTO-Stage 1)} \\ S_{interstage}\ 22.94\ D_{body}^{-0.6751} & \text{(TSTO-Stage 2)} \end{cases} \tag{6.19}$$

$$M_{fwd\ skirt} = \begin{cases} S_{fwd\ skirt}\ 37.35\ D_{body}^{-0.6722} & \text{(SSTO)} \\ S_{fwd\ skirt}\ 38.70\ D_{body}^{-0.6722} & \text{(TSTO-Stage 1)} \\ S_{fwd\ skirt}\ 15.46\ D_{body}^{-0.5210} & \text{(TSTO-Stage 2)} \end{cases} \tag{6.20}$$

$$M_{fwd\ skirt} = \begin{cases} S_{fwd\ skirt}\ 37.35\ D_{body}^{-0.6722} & \text{(SSTO)} \\ S_{fwd\ skirt}\ 38.70\ D_{body}^{-0.6722} & \text{(TSTO-Stage 1)} \\ S_{fwd\ skirt}\ 15.46\ D_{body}^{-0.5210} & \text{(TSTO-Stage 2)} \end{cases} \tag{6.21}$$

$$M_{eng\ comp} = \begin{cases} S_{eng\ comp} 31.66 D_{body}^{-0.5498} & \text{(SSTO)} \\ S_{eng\ comp} 32.48 D_{body}^{-0.5498} & \text{(TSTO-Stage 1)} \\ S_{eng\ comp} 15.97 D_{body}^{-0.4676} & \text{(TSTO-Stage 2)} \end{cases} \tag{6.22}$$

The mass of aft skirts is estimated from Equation 6.23, which shows that its mass is function of the maximum dynamic pressure and the body diameter.

$$\begin{aligned} M_{aft\ skirt} = S_{aft\ skirt} \big[& 2.499 \cdot 10^{-4} Q_{max} + 1.7008 + \\ & \left(3.695 \cdot 10^{-5} Q_{max} - 3.252e-3\right) D_{body} \big] \end{aligned} \tag{6.23}$$

Finally, the stage attachment and thrust structure masses are estimated according to Equations 6.24 and 6.25. The stage attachment mass is thus function of the total mass of the boosters, and the thrust structure is a function of the total thrust of the engines.

$$M_{attch} = 0.00148 M_{booster} N_{booster} \tag{6.24}$$

$$M_{thrust} = 7.995 \cdot 10^{-4} (T_{vac} N_{eng})^{1.0687} \tag{6.25}$$

All the subsystem masses described in Equation 6.14 are described in Equations 6.15 to 6.25 to compute the total body weight of liquid rocket vehicles. The next section describes the procedure to compute the body weight of solid rocket stages and boosters.

Solid Rocket Vehicles For solid rocket stages or boosters, the body mass is computed by estimating the equations developed by Brothers, 1999[9] and Rohrschneider[82]. The design parameters required to estimate the body mass of solid rocket stages are defined in Table 6.6.1.

The overall body mass of solid rocket vehicles is computed from Equation 6.26, which implies that the stage is expendable.

$$M_{body} = M_{case} + M_{nozzle} + M_{aft\ skirt} + M_{fwd\ skirt} + M_{interstage} + M_{nose}; \tag{6.26}$$

Each of the six mass components of Equation 6.26 are explained in more detail below. First, the case mass, being usually the heaviest component for solid rocket vehicles, is estimated by the MER depicted in Equation 6.27. It is function of the case volume, as well as the its diameter.

$$
\begin{aligned}
M_{case} = \; & V_{case}\left[\left(1.07 \cdot 10^{-7} D + 9.1014 \cdot 10^{-3}\right) 946 - \left(8.537 \cdot 10^{-4} D - 6.483 \cdot 10^{-3}\right)\right] \\
& - \left(9.1677 \cdot 10^{-7} \left(12D\right)^{3.008} 946 + 53.65 - 4.128D\right)
\end{aligned}
\tag{6.27}
$$

Second, the nozzle mass is estimated from Equation 6.28, which is function of the body diameter, the propellant mass, and the motor specific impulse.

$$
M_{nozzle} = 1.085 \cdot 10^{-6} \left(12 \cdot D\right)^3 946 + 4.142 \cdot 10^{-5} Isp_{vac} M_{prop} \tag{6.28}
$$

Third, the forward and aft skirts are estimated by Equations 6.29 and 6.30, respectively. The forward skirt mass varies as a function of the stage number, the skirt area, and body diameter. The aft skirt mass is mainly function of its surface area, the maximum dynamic pressure, and the body diameter.

$$
M_{fwd\ skirt} = \begin{cases} S_{fwd\ skirt}\ 37.35\ D^{-0.6722} & \text{(Booster)} \\ S_{fwd\ skirt}\ 38.70\ D^{-0.6722} & \text{(Stage 1)} \\ S_{fwd\ skirt}\ 15.46\ D^{-0.5210} & \text{(Stage 2)} \end{cases} \tag{6.29}
$$

Finally, the aft skirt mass is computed from Equation 6.30 and is thus function of the skirt wetted area, the maximum dynamic pressure and the body diameter.

$$
\begin{aligned}
M_{aft\ skirt} = S_{aft\ skirt} \big(& 2.499 \cdot 10^{-4} Q_{max} + 1.7008 + \\
& \left(3.695 \cdot 10^{-5} Q_{max} - 3.252 \cdot 10^{-3}\right) D\big)
\end{aligned}
\tag{6.30}
$$

Similar to liquid rocket stages, the interstage mass differs if the stage is used as a booster, a core first stage or a second stage, as depicted in Equation 6.31.

$$
M_{interstage} = \begin{cases} S_{interstage}\ 17.92\ D^{-0.4856} & \text{(Booster)} \\ S_{interstage}\ 18.57\ D^{-0.4856} & \text{(Stage 1 of 2)} \\ S_{interstage}\ 22.94\ D^{-0.6751} & \text{(Stage 2 of 2)} \end{cases} \tag{6.31}
$$

For solid rocket boosters, the nose mass is estimated using Equation 6.32 for a conical nose. The equation that the nose mass is function of the maximum dynamic pressure, the nose cone angle, the body diameter, and the nose surface area.

$$\begin{aligned} M_{nose} = \quad & S_{nose} \Big[\left(14.31 - 0.003462 Q_{max}\right) \lambda^{\left(1.034 \cdot 10^{-4} Q_{max} - 0.5878\right)} \\ & + \left(\left(6.864 \cdot 10^{-4} - 6.1 \cdot 10^{-9} Q_{max}\right) \lambda + \left(4.385 \cdot 10^{-5} Q_{max} - 0.037\right) \right) D \Big] \\ & + S_{nose} \left(\left(6.656 \cdot 10^{-4} \lambda - 1.0787 \cdot 10^{-3}\right) D + 2.8888 - 0.026777 \lambda \right) \end{aligned} \tag{6.32}$$

In summary, Equations 6.27 to 6.32 are used to compute the total body mass of solid rocket stages or boosters. The next section describes how the propulsion system mass is computed.

Table 6.13: Description of the parameters used for computing the body weight of solid rocket vehicles

Parameter	Description
α	Nose cone angle, deg
D	Diameter, ft
Isp_{vac}	Specific impulse in vacuum, s
M_{prop}	Propellant mass, lbm
Q_{max}	Maximum dynamic pressure, psf
$S_{aft\ skirt}$	Aft skirt wetted area, ft^2
$S_{fwd\ skirt}$	Forward skirt wetted area, ft^2
$S_{interstage}$	Interstage wetted area, ft^2
S_{nose}	Nose wetted area, ft^2
V_{case}	Solid case volume, ft^3

2 - Main Propulsion Group

This section describes the MERs for computing the main propulsion group mass. First, the design parameters required to compute this subsystem mass are shown in Table 6.6.1.

Table 6.14: Description of the parameters used for computing the main propulsion group mass

Parameter	Description
F_{ull}	Ullage fraction (0.02)
$\dot{m}_p$	Propellant mass flow rate, lbm/s
M_{ei}	Engines installation mass, lbm
$M_{engines}$	Engines mass, lbm
M_{es}	Engine subsystems mass
M_{feed}	Propellant feed system mass, lbm
M_{tvc}	Thrust vector control system mass, lbm
N_{eng}	Number of engines
T_{vac}	Vacuum thrust, lbf
P_{tank}	Propellant tank pressure, psf
V	Body volume, ft^3

As depicted in Equation 6.33, the mass of the main propulsion group includes the engines mass ($M_{engines}$), the engines installation mass(M_{ei}), the engines subsystem mass(M_{es}), the trust vectoring control system mass(M_{tvc}), the purge system mass (M_{purge}), the propellant feed system mass (M_{feed}), and the pressurization system mass (M_{ps}).

$$M_{main\ prop} = M_{engines} + M_{ei} + M_{es} + M_{tvc} + M_{purge} + M_{feed} + M_{ps} \tag{6.33}$$

The engines mass is computed from Equation 6.34, which differentiates between the two propellant types: 1)LOX/LH2 and 2) LOX/RP. In both cases, the vacuum thrust, and the number of engines are the two regression parameters.

$$M_{engines} = \begin{array}{ll} \frac{T_{vac}N_{eng}}{min(75,5.11ln(T_{vac}N_{eng})+4.2)} & (LOX/LH2) \\ \frac{T_{vac}N_{eng}}{min(104.4,26.04ln(T_{vac}N_{eng})-207)} & (LOX/RP) \end{array} \tag{6.34}$$

The engine installation, engine subsystem and thrust vector control MERs are described by Equations 6.35, 6.36, and 6.37, respectively. These three subsystems are function of the total vacuum thrust supplied by the engines.

$$M_{ei} = 5.6e-4 * T_{vac}N_{eng} \tag{6.35}$$

$$M_{es} = 5.6e-4 T_{vac}N_{eng} \tag{6.36}$$

$$M_{tvc} = 0.001185 T_{vac}N_{eng} \tag{6.37}$$

The propellant purge and feed subsystems are estimated from Equations 6.38 and 6.39, respectively. The purge system is function of the total propellant volume, whereas the feed system is function of the total propellant mass flow rate.

$$M_{purge} = 0.053V \tag{6.38}$$

$$M_{feed} = 2.197\dot{m}_p \tag{6.39}$$

Finally, the pressurization system mass is computed from Equation 6.40. Three options are available for the different propellant types : 1) pressure-fed, 2) cryogenic, and 3) storable. The pressure-fed options assumes that the pressurization gas is cold nitrogen (N2).[82]

$$M_{ps} = \begin{cases} 0.55\left(1.302 + 0.99P_{tank}\right)V^{\left(0.8647P_{tank}^{0.01645}\right)} & (pressurefed) \\ 0.192\dot{m}_p & (cryogenic) \\ 50 + 0.192\dot{m}_p + \frac{F_{ull}}{26}0.18V & (storable) \end{cases} \tag{6.40}$$

3 - Reaction Control System (RCS) Group

The RCS mass (M_{rcs}) is computed from Equation 6.49, where the RCS can use either cryogenic or storable propellants. The group mass is function of the empty mass (M_e) and the stage length (L).

$$M_{rcs} = \begin{cases} 1.36e-4 M_e L & \texttt{(storable)} \\ 1.51e-4 M_e L & \texttt{(cryogenic)} \end{cases} \tag{6.41}$$

4 - Orbital Maneuver System (OMS) Group

The orbital maneuver system mass (M_{OMS}) is computed from Equation 6.42, which shows its proportionality with the vehicle empty mass (M_e).

$$M_{OMS} = 0.0121 M_e \tag{6.42}$$

5 - Primary Power Group

The primary power group mass is estimated based on the power requirements of the hydraulic system and the avionics. It is estimated by Equation 6.43, where M_{prop}, T_{vac}, N_{eng}, and $M_{avionics}$, are the propellant mass, the vacuum thrust, the number of engines, and the avionics mass, respectively.

$$W_{pow} = 793 + 506 \left(\frac{M_{prop}}{1.6 \cdot 10^6} \right) + 9.7 \cdot 10^{-5} T_{vac} N_{eng} + 0.405 M_{avionics} \tag{6.43}$$

6 - Electrical Conversion and Distribution (ECD) Group

The ECD group mass is estimated from Equation 6.44, where M_{empty} is the empty mass.

$$W_{ECD} = 0.062 M_{empty} \tag{6.44}$$

7 - Hydraulic Systems Group

The hydraulic systems group computes the system mass thanks to Equation 6.45, where T_{vac} represents the total vacuum thrust of the stage.

$$W_{hyd} = 3e - 4T_{vac} \tag{6.45}$$

8 - Avionics Group

The avionics mass is taken as a constant, as displayed in Equation 6.46, where the different avionics system masses are shown.

$$M_{av} = \begin{cases} 6564 & \texttt{(Space Shuttle)} \\ 670 & \texttt{(CaLV Stage 1)} \\ 430 & \texttt{(CaLV Stage 2)} \end{cases} \tag{6.46}$$

9 - Residual Propellants Group

The residual propellants mass is estimated from Equation 6.47 as a function of the total ascent propellant mass.

$$M_{res} = 0.05 M_{propasc}^{0.79} \tag{6.47}$$

10 - OMS and RCS Reserve Propellants

The OMS and RCS reserve propellants are function of the ΔV requirements, as displayed in Equation 6.48

$$M_{omsrcsres} = M_{empty}\left[exp\left(\frac{0.005 \cdot \Delta V_{oms}}{Isp_{oms}g}\right) + exp\left(\frac{0.005 \cdot \Delta V_{rcs}}{Isp_{oms}g}\right) - 2\right] \tag{6.48}$$

$$\begin{aligned} M_{OMSRCSres} &= \text{OMS and RCS reserve propellant mass, lbm} \\ M_{land} &= \text{Vehicle mass at landing, lbm} \\ \Delta V_{oms} &= \text{Total velocity change using OMS engines} \\ \Delta V_{rcs} &= \text{Total velocity change using RCS engines} \\ Isp_{oms} &= \text{Specific impulse of OMS engines} \\ Isp_{rcs} &= \text{Specific impulse of RCS engines} \end{aligned}$$

11 - RCS Entry Propellants Group

The RCS propellant mass is calculated from the system ΔV requirements and uses the basic rocket equation, as shown by Equation 6.49

$$M_{rcsentry} = M_{land}\left[exp\left(\frac{0.005\Delta V_{rcs\ entry}}{Isp_{rcs} \cdot g}\right) - 1\right] \tag{6.49}$$

$$\begin{aligned} M_{rcsentry} &= \text{Reentry RCS propellant mass, lbm} \\ M_{land} &= \text{Vehicle mass at landing, lbm} \\ \Delta V_{rcsentry} &= \text{Total velocity change during entry using RCS engines} \\ Isp_{rcs} &= \text{Specific impulse of RCS engines} \end{aligned}$$

12 - OMS and RCS On-Orbit Propellants Group

The on-orbit propellant represent the propellant burned by the OMS and RCS

for orbital maneuvers. These propellant masses are estimated by the basic rocket equation. This model is summarized by Equation 6.50,

$$M_{OMSRCSprop} = M_{empty}\left[exp\left(\frac{\Delta V_{oms}}{Isp_{oms}g}\right) + exp\left(\frac{\Delta V_{rcs}}{Isp_{oms}g}\right) - 2\right] \tag{6.50}$$

where

$$\begin{array}{rcl} M_{empty} & = & \text{Vehicle mass at landing, lbm} \\ \Delta V_{oms} & = & \text{Total velocity change using OMS engines} \\ \Delta V_{rcs} & = & \text{Total velocity change using RCS engines} \\ Isp_{oms} & = & \text{Specific impulse of OMS engines} \\ Isp_{rcs} & = & \text{Specific impulse of RCS engines} \end{array}$$

Table 6.15 shows the effect of the propellant type on the OMS or RCS specific impulse. These values are used in the later investigation of the OMS and RCS types.

13 - Ascent Reserve Propellants Group

The reserve propellant mass is the subject of this mass group. The MER used for its estimation is presented by Equation 6.51, where $M_{prop\ asc}$ represents the ascent propellant mass of the stage.

$$M_{ascent\ res.} = 0.005 M_{prop\ asc} \tag{6.51}$$

14 - Inflight Losses and Vents Group

The propellant mass losses occurred during the flight and through tank vents is

Table 6.15: Specific impulse used for the OMS and RCS

Type	Cryogenic	Storable
OMS	440 s	313 s
RCS	398 s	289 s

estimated by Equation 6.52, where $M_{prop\ asc}$ represents the ascent propellant mass of the stage.

$$M_{flloss} = 0.0043 M_{prop.\ asc.} \tag{6.52}$$

$$\begin{aligned} M_{flloss} &= \text{Ascent reserve propellant mass, lbm} \\ M_{propasc} &= \text{Ascent propellant mass, lbm} \end{aligned}$$

15 - Startup Losses Group

The startup losses represent the propellant losses occurring before the launch vehicle liftoff. They are estimated by Equation 6.53, where $M_{prop\ asc}$ represents the ascent propellant mass of the stage.

$$M_{startup\ losses} = 0.01 M_{prop.\ asc.} \tag{6.53}$$

These 15 mass groups are used to estimate the total gross weight of each stage of the vehicle, given the required propellant mass. The next section presents the verification results of the mass estimating relationships.

6.6.2 Verification of the Weight Model

The MERs presented in the previous section are verified using either existing launch vehicles when the data is available, or launch vehicle designs when the data is non-existent. This section presents the comparison between the values obtained from the weight estimating equations and the actual vehicle measurements.

Space Shuttle

The Space Shuttle orbiter is used to verify the various empty weight subsystems. The MERs presented in the previous section are directly used with the inputs presented in Table 6.16 to verify the mass model. In other words, no calibration factor is used to adjust the error, which results in some large errors for few subsystems, as shown in the next paragraphs. It is recalled, however, that the focus of this thesis is to demonstrate the concept and technology down-selection methodology.

Table 6.17 shows the mass comparison between the RASAC-3's model and the actual masses per subsystem for the Space Shuttle. The wing, tail, landing gear, and control surface subsystems are omitted because they are not considered the implementation case presented in the next chapter. The results show a good accuracy for predicting the various Space Shuttle subsystems. The highest error is a little above 10%, and most of the other subsystem such as the body, the main propulsion, and the Electrical Conversion and Distribution (ECD) are below this 10% mark.

Table 6.18 shows the good accuracy of the model for predicting the tank dry mass, as seen by the small amount of error, except for the RCS group. Moreover, the mass model exhibits conservatism regarding the propellant mass for reserve and losses. The MERs for the reserve, inflight losses & vents, and startup losses are, however, consistent with the literature standards[82], being equal to 0.5%, 0.43%, and 0.2% of the ascent propellants, respectively.

Table 6.16: Space Shuttle inputs used for the weight model verification

Parameter	Value	Units
Mission		
Maximum dynamic pressure	650	psf
Number of crews	7	-
Mission duration	7	days
Δ V OMS	700	ft/s
Δ V RCS	200	ft/s
Orbiter		
Vehicle length	107	ft
Body wetted area	6184	ft^2
Fuselage wetted area	6636	ft^2
Body Planform	1350	ft^2
Body width	17	ft
Insertion mass	268376	lbm
External tank		
Vehicle length	153.8	ft
Body width	27.583	ft
Intertank wetted area	1949.75	ft^2
Oxidizer tank wetted area	4527	ft^2
Fuel tank wetted area	6578	ft^2
Oxidizer tank volume	53518	ft^3
Fuel tank volume	19563	ft^3
Oxidizer-to-fuel ratio	6	-
Mass propellant ascent	1568428	lbm
Main engines		
Vacuum thrust	512,000	lbf
Number of engines	3	-

Table 6.17: Comparison between the predicted and the actual values of the Space Shuttle Orbiter per subsystem mass group

Mass group	RASAC-3, lbm	Actual, lbm	Error
Body	45768	43922	4.2%
Main propulsion	28396	31006	-8.4%
RCS	3466	3142	10.3%
OMS	3247	3041	6.8%
Primary power	3909	3912	-0.1%
ECD	10376	10469	-0.9%
Hydraulics	1807	1853	-2.5%
Avionics	6564	6557	0.1%
Env. control	5303	5304	0.0%
Personnel equipment	1703	1836	-7.2%
OMS/RCS On-orbit prop.	22297	22138	0.7%

Table 6.18: Comparison between the predicted and the actual values of the Space Shuttle external tank per subsystem mass group

Mass group	RASAC-3, lbm	Actual, lbm	Error
Body	67235	66000	1.90%
Residuals	3921	4621	-15.2%
Ascent reserves	7842	2344	234.6%
Inflight losses & vents	6744	3680	83.3%
Startup losses	3137	4015	-21.9%

6.7 Cost Modeling

This section presents the cost estimating approach modeled in RASAC-3. The approach is based on Cost Estimating Relationships (CER's), which take the vehicle subsystem weights as the independent parameters. By its nature, it is a model that shows the cost gross trends and sensitivity of the different concepts rather than an absolute model for which engineering decisions should be taken. In other words, it is used to compare alternatives between one another during the concept exploration cost.

The cost model is based on the response surface equations obtained from the NASA Air Force Cost Model (NAFCOM) [51] for computing the cost of liquid rocket stages, and on Transcost [60] for computing the cost of solid rocket stages. The cost model computes the Theoretical First Unit (TFU) and Design Development Testing and Engineering (DDT&E) costs. The CERs are formulated in a form similar to Equation 6.54, where a and b are the statistically-regressed coefficients, K a complexity factor, and M is the mass of the system. The next sections describe the various CERs. Table 6.7 describes the parameters used for computing the cost estimating relationships. These variables are used throughout this section.

$$Cost = KaM^b \tag{6.54}$$

6.7.1 Solid Rocket Boosters

The development and production costs for solid rocket boosters are estimated by Equations 6.55 and 6.56, respectively, where f_1 is the development standard factor, f_2 is the technical quality factor, f_3 is the team experience factor, M_{dry} is the dry mass, and M_{net} the net mass in kg. The net mass is equal to the initial mass of the vehicle minus the payload and ascent propellant, and the net mass fraction is the ratio of the net mass to propellant mass ($NMF = M_{net}/M_{prop}$).

Table 6.19: Description of the parameters used for computing the cost estimating relationships

Parameter	Description
f_1	Development standard factor
f_2	Technical quality factor
f_3	Team experience factor
$margin$	Cost margin
A, B, F	Regression coefficients
C_d	Development cost
$C_{a.\ d.}$	Airframe development cost, \$M
$C_{e.\ d.}$	Engine development cost, \$M
$C_{a.\ tfu}$	Airframe production cost, \$M
$C_{e.\ tfu}$	Engine production cost, \$M
C_i	Cost of subsystem i
C_p	Production cost
$C_{sub.\ a.\ tfu}$	Airframe production cost subtotal, \$M
$C_{sub.\ e.\ tfu}$	Engine production cost subtotal, \$M
$C_{sub.\ a.\ d.}$	Airframe development cost subtotal, \$M
$C_{sub.\ e.\ d.}$	Engine development cost subtotal, \$M
C_{GSE}	Ground support equipment cost, \$M
C_{IACO}	Integration, assembly, and checkout cost, \$M
C_{STH}	System test hardware cost, \$M
C_{PM}	Program management cost, \$M
$C_{SE\&I}$	System engineering and integration cost, \$M
C_{STO}	System test operations cost, \$M
F_{GSE}	Ground support equipment cost regression coefficient
F_{IACO}	Integration, assembly, and checkout cost regression coefficient
F_{STH}	System test hardware cost regression coefficient
F_{PM}	Program management cost regression coefficient
$F_{SE\&I}$	System engineering and integration cost regression coefficient
F_{STO}	System test operations cost regression coefficient
M_{dry}	Dry mass, lbm
M_{net}	Engines mass, lbm
M_{prop}	Propellant mass, lbm
M_i	Mass of subsystem i, lbm

$$Cd = f_1 f_2 f_3 \left(4.9 \left(M_{net}\right)^{0.68}\right) \tag{6.55}$$

$$Cp = f_4 n \left(2.42 \left(M_{dry}\right)^{0.395}\right) \tag{6.56}$$

6.7.2 Expendable Liquid Rocket Vehicles

The development and production costs of expendable launch vehicle stages is computed using a surrogate model from NAFCOM [51]. The development and production costs are computed in two steps. The first step consists of calculating the cost of the vehicle subsystems described in the first column of Table 6.20.

Table 6.20: Expendable launch vehicle subsystem cost groups

Subsystem group
Body
Main propulsion
RCS propulsion
OMS propulsion
Primary power
Electrical conversion & dist.
Hydraulic systems
Avionics
Environmental control

The cost of subsystem i is computed with the help of Equation 6.57, where C_i and M_i represent the cost and weight of subsystem i, respectively, K, the complexity factor, A, and B, two cost regression coefficients.The cost regression coefficients (A and B), and the complexity factors vary for each subsystem and vehicle type. For expendable launch vehicle stages, though, the systems showed in Table 6.20 are used to compute the total development and production costs. The cost regression coefficients are regressed from NAFCOM [51].

$$C_i = KAM_i^B \tag{6.57}$$

The second step consists of summing the cost of each subsystem and then computing the cost of processing, as shown by Equations 6.59, 6.66, 6.73, and 6.69. The processing, assembly, and integration costs are computed to determine the total TFU and development costs. This approach then lead to the computation of the total airframe and engine costs, as displayed in Equation 6.58.

$$C_{sub.\ a.\ d.} = \sum_{i=1}^{i=n} C_i \tag{6.58}$$

The Design Development, Testing and Engineering (DDT&E) cost is divided into the airframe and engine development costs. First, the airframe total development cost is expressed by Equation 6.59. The cost coefficients (F) for the DDT&E cost computation are regressed from NAFCOM [51].

$$\begin{aligned} C_{a.\ d.} &= (C_{STH} + C_{IACO} + C_{STO} + C_{GSE} + \\ &\quad C_{SE\&I} + C_{sub.\ a.\ d.})\,(1 + margin) \end{aligned} \tag{6.59}$$

where

$$C_{STH} = F_{STH} \cdot C_{sub.\ a.\ d.} \tag{6.60}$$

$$C_{IACO} = F_{IACO} \cdot C_{STH} \tag{6.61}$$

$$C_{STO} = F_{STO}\,(C_{STH} + C_{IACO}) \tag{6.62}$$

$$C_{GSE} = F_{GSE}\,(C_{STO} + C_{sub.\ a.\ d.}) \tag{6.63}$$

$$C_{SE\&I} = F_{SE\&I}(C_{STH} + C_{IACO} + C_{STO} + C_{GSE} + C_{sub.\ a.\ d.}) \tag{6.64}$$

$$C_{PM} = F_{PM}\,(C_{STH} + C_{IACO} + C_{STO} + C_{GSE} + C_{SE\&I} + C_{sub.\ a.\ d.}) \tag{6.65}$$

Similar to the airframe, the engine total development cost is estimated from Equation 6.66,

$$C_{\ e.\ d.} = (C_{SE\&I} + C_{PM} + C_{sub.\ e.\ d.})\,(1 + costmargin) \tag{6.66}$$

where $C_{sub.\ e.\ d.}$ is the sum of the engine development subsystem costs, and

$$C_{SE\&I} = F_{SE\&I} C_{sub.\ e.\ d.} \tag{6.67}$$

$$C_{PM} = F_{PM} \left(C_{SE\&I} + C_{sub.\ e.\ d.} \right) \tag{6.68}$$

The values assigned for the factors described in the above equation are taken from NAFCOM [51]. Similar to the DDT&E cost, the airframe TFU cost is estimated from Equation 6.69,

$$C_{a.\ tfu} = \left(C_{IACO} + C_{SE\&I} + C_{PM} + C_{sub.\ a.tfu} \right) (1 + margin) \tag{6.69}$$

where,

$$C_{IACO} = F_{IACO} C_{sub.\ a.\ t.} \tag{6.70}$$

$$C_{SE\&I} = F_{SE\&I} * \left(C_{IACO} + C_{sub.\ a.tfu} \right) \tag{6.71}$$

$$C_{PM} = F_{PM} \left(C_{IACO} + C_{SE\&I} + C_{sub.\ a.tfu} \right) \tag{6.72}$$

Different processing cost factors coefficients [51] are used to compute the above equations. Finally, the engine TFU cost is estimated from Equation 6.73. This closes the loop for the cost estimation of the stage. Therefore, for any expendable stage the TFU and DDT&E costs can be estimated from the weight breakdown of each subsystem.

$$C_{e.\ tfu} = \left(C_{SE\&I} + C_{PM} + C_{sub.\ e.\ tfu} \right) (1 + costmargin) \tag{6.73}$$

where $C_{sub.\ e.\ t.}$ is the sum of the subsystem engine TFU costs, and

$$C_{SE\&I} = F_{SE\&I} C_{sub.\ e.\ tfu} \tag{6.74}$$

$$C_{PM} = F_{PM} \left(C_{SE\&I} + C_{sub.\ e.\ tfu} \right) \tag{6.75}$$

6.8 Sizing and Synthesis

6.8.1 Process Description

The sizing and synthesis of the launch vehicle is performed as depicted by Algorithm 7. The first step consists of sizing the engine for the specified stages gross masses; this approach is specified in Section 6.5. Then, the second step consists of computing the trajectory simulation to get the required propellant masses for each stage. The trajectory simulation model is presented in Section 6.3. This required propellant mass is then used to compute the geometry of the vehicle (length, diameter). This represents the third step. Once the dimensions and propellant required are known, the subsystem masses is computed as described in Section 6.6, which is the fourth step of the sizing and synthesis process. These subsystem masses are then used to determine the new vehicle gross weight. This process is repeated until the vehicle Gross Liftoff Weight (GLOW) converges.

while *Gross liftoff weight* $\neq$ *Gross liftoff weight (old)* **do**
    Size engines
    Compute trajectory
    Compute geometry
    Compute subsystem mass
    Compute new gross masses
end

Algorithm 7: Sizing and synthesis structure

Table 6.21: Mission inputs for the CaLV verification

Parameter	Value
Maximum dynamic pressure, psf	600
Payload mass, lbs	133,703
Fairing mass, lbs	10,522
Final relative velocity, ft/s	24,365
Final altitude, ft	475,870
Final flight path angle, deg	1.0

6.8.2 Verification

The entire sizing and synthesis process of RASAC-3 is verified and presented below. The vehicle used for its verification is the CaLV, since it is the launch vehicle used for the demonstration of the concept and technology down-selection methodology presented in the next chapter. The goal of this section is to show that the RASAC-3, the launch vehicle sizing and synthesis code, can predict the CaLV with a sufficient level of fidelity, and that the code can be used in the demonstration of the methodology.

For the verification of the CaLV, the mission parameter presented in Table 6.21 were used, representing a trajectory of a 30x160 nmi orbit (injected at 78.3 nmi), as stipulated in the ESAS report.[1] The orbit and the launch pad are assumed to be at 28.5° of inclination.

The boosters modeled are five-segment solid rocket boosters. The first and second stages use LOX/LH2 liquid rocket engines, similar to SSME (core first stage) and J2-X (core second stage) engines. The remaining engine parameters such as the chamber pressure, propulsive efficiency, and expansion ratio are described in Table 6.22.

Using the input parameters mentioned above, the CaLV performance is computed with RASAC-3 and Tables 6.23 to 6.26 show the results obtain from the simulation, where the mass ratio required are obtained directly from the trajectory module, not the response surface equations.

Table 6.22: Propulsion and geometry inputs for the CaLV

Parameter	Boosters	Stage 1	Stage 2
Thrust-to-weight ratio	1.176	0.327	0.835
Chamber pressure, psi	978	3617	1283
Expansion ratio	6.5	77.5	28.0
Propulsive efficiency	0.974	0.977	0.997
Burnout velocity	3727	17000	24,365
Reference area, ft^2	826.48	826.48	593.96
Aft skirt length, ft	2.29	7.00	8.00
Tank clearance, ft	-	5.00	6.00
Interstage length, ft	-	5.00	-
Forward skirt length, ft	10.87	-	6.00
Engine compartment length, ft	0.00	14.00	10.00

Table 6.23: CaLV required mass ratio from the trajectory module

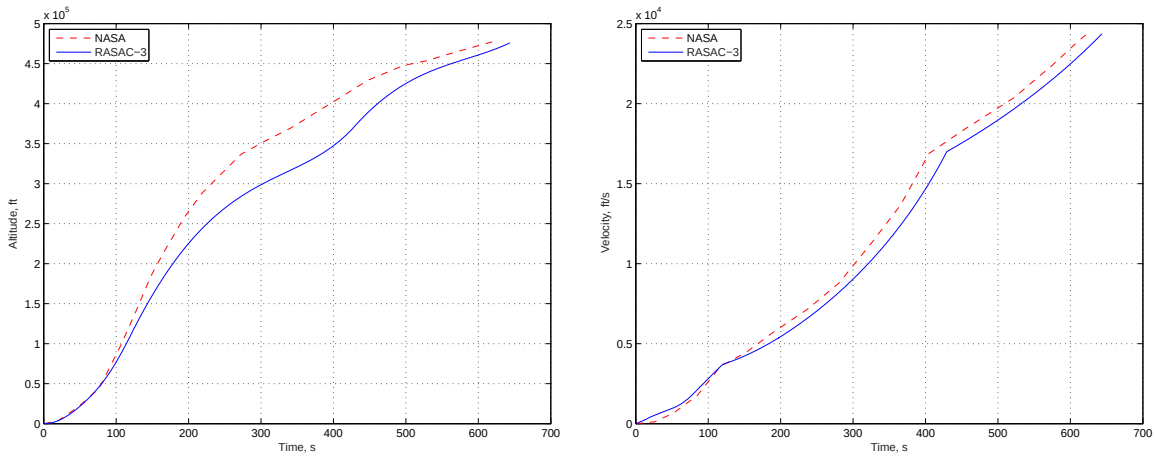
	ESAS	RASAC-3	Error
Booster mass ratio*	1.82	1.778	-2.33 %
Stage 1 mass ratio	3.59	3.576	-0.4 %
Stage 2 mass ratio	3.511	3.405	-3.0 %

* First stage (core) gross weight used for the payload weight

First, the mass ratio requirements computed from the trajectory simulation module are presented in Table 6.23. It shows that the performance of the launch vehicle computed from RASAC-3 is in good agreement with the results obtained from NASA's ESAS report.[1] The trajectory is also compared in Figure 6.8.2, which shows the altitude and velocity variation as function of time for the entire trajectory.

Second, Table 6.24, 6.25, and 6.26 describe the mass breakdown for the entire launch vehicle, the first stage, and the second stage, respectively. These tables show that the gross masses and ascent propellant masses of each stage computed by RASAC-3 are in agreement with those from the ESAS study. The gross masses of all three stages have error below 4 %, whereas the stage gross and propellant mass errors are sufficiently low to insure good performance estimation.

The CaLV first and second stages mass breakdown are also presented in Tables 6.25



(a) (b)

Figure 6.10: Altitude and velocity variation as a function of time for the final sizing and synthesis process

Table 6.24: Strap-on boosters actual vs predicted mass comparison

	ESAS	RASAC-3	Error
Dry mass, lbs	221234	223601	1.1 %
Gross mass, lbs	1656140	1639822	-1.0 %
Ascent propellant mass, lbs	1434906	1409174	-1.8 %

Table 6.25: CaLV First stage (core) mass breakdown comparison

Group	ESAS	RASAC-3	Error
Body mass, lbs	106754	108391	1.5 %
Propulsion mass, lbs	58015	57130	-1.5 %
Power mass, lbs	4726	4549	-3.7 %
Avionics mass, lbs	670	670	0.0 %
Growth mass, lbs	14413	14318	-0.7 %
Residuals mass, lbs	16676	16712	0.2 %
Reserves mass, lbs	3323	3332	0.3 %
Inflight losses mass, lbs	262	263	0.3 %
Propellant ascent mass, lbs	2215385	2221607	0.3 %
Dry mass, lbs	194997	193293	-0.9 %
Gross mass, lbs	2430894	2522618	-3.8 %

Table 6.26: CaLV second stage (core) mass breakdown comparison

Group	ESAS	RASAC-3	Error
Body mass, lbs	21998	21901	-0.4 %
Propulsion mass, lbs	12642	12583	-0.5 %
Power mass, lbs	1817	1749	-3.7 %
Avionics mass, lbs	430	430	0.0 %
Growth mass, lbs	3599	3510	-2.5 %
Residuals mass, lbs	5309	5319	0.2 %
Reserves mass, lbs	628	629	0.2 %
Inflight losses mass, lbs	59	59	0.1 %
Propellant ascent mass, lbs	457884	459233	0.3 %
Dry mass, lbs	42645	42686	0.1 %
Gross mass, lbs	506576	520728	-2.8 %

and 6.26. Overall, the error from the various subsystem mass estimation are small and the the dry and gross masses from the stage are fairly accurate as discussed in the previous paragraph. This low level of error is related to the calibration of the MERs, presented in Section 6.6, with the data obtained from the ESAS report [1] on the CaLV.

Table 6.26 shows the mass breakdown comparison for the CaLV core second stage. The body and propulsion system masses, the main dry mass contributors, are in good agreement with the results from the ESAS report [1]. A complete breakdown of the core first and second stage masses is presented in Appendix C.

CHAPTER VII

IMPLEMENTATION & RESULTS: LAUNCH VEHICLE CONCEPT AND TECHNOLOGY SELECTION

As stated in Chapter 1, the purpose of this thesis is to develop a concept and technology down-selection methodology within aerospace architecture studies. This chapter presents a demonstration of this methodology by performing a concept and technology down-selection on the CaLV launch vehicle architecture. The next sections show how this methodology can bring higher fidelity, and help design engineers to down-select from a set of concepts and technologies. The sections also show how the hierarchical and incompatibility problems discussed in Chapter 2 can be alleviated by using Ant Colony Optimization to down-select the concept and technology alternatives.

7.1 Application of the Concept and Technology Exploration Methodology

7.1.1 Step 1: Define the Requirements

This problem consists of selecting the best launch vehicle architecture to deliver the Lunar Surface Access Module (LSAM) to a 30x160 nmi orbit at 28.5 deg inclination (injected at 78.3 nmi). Moreover, the vehicle must inject the CEV and LSAM to a translunar trajectory, for which 10,334 fps of velocity change (ΔV) is required. A summary of the mission requirements is presented in Table 7.1. These requirements are taken from the NASA architecture study [1] but the concept exploration study presented in Chapter 7 is different from the ESAS, as the engines are rubberized and

Table 7.1: Mission inputs for the CLV verification

Parameter	Value
Payload mass, lbs	59,898
Final relative velocity, ft/s	24,368
Final altitude, ft	475,870
Final flight path angle, deg	0.594
Orbit inclination, deg	28.5
Translunar injection ΔV, ft/s	10,334

the strap-on booster masses and thrust are varied.

The figure of merit for this launch vehicle design is an equally-weighted overall evaluation criterion, composed of the Design, Development, Testing and Engineering (DDTE), and the Theoretical First Unit (TFU) costs. The overall evaluation criterion is shown in Equation 7.1, where the subscript ref is a reference value for normalization. The values used for normalization of the are those obtained from the baseline CaLV architecture concept, that is TFU_{ref} = \$2,912 M, and $DDTE_{ref}$ = \$17,374 M.

$$F = \frac{DDTE_{ref}}{DDTE} + \frac{TFU_{ref}}{TFU} \tag{7.1}$$

The author is also aware that using an OEC for optimization can be inferior to other Multi-Attribute Decision Making approaches such as Pareto optimality [62], or TOPSIS [57], where the OEC can lead to sub-optimal solutions[63]. However, for a real-world application the design engineer can use any of the Multi-Attribute Decision Making techniques as a replacement for the OEC to determine the quality of a design from its attributes. The final answer is likely to change but the process still applicable.

The current cost model, as presented in Section 6.7 uses the same regression coefficients for all the concepts and technologies considered in the morphological matrix. Moreover, both the TFU and the DDT&E costs are function of the vehicle dry mass, and the TFU cost is a function of the DDT&E cost. This relationship between the

Table 7.2: Concepts for the launch vehicle example

Concepts
Single and two stage vehicle
Solid and liquid rocket boosters
LOX/LH2, LOX/RP1 and P-BAN
Pump-fed and Pressure-fed rocket engines
Storable and cryogenic OMS/RCS systems

two OEC cost components included thus implies that they are not entirely independent and their weight may not even effect the outcome of the optimization. The goal of this chapter, however, is to demonstrate the proposed concept and technology selection methodology, and to show how the methodology could consider more than one selection criterion.

In addition to the OEC, an exterior penalty function is applied to launch vehicle concepts exceeding 8,000,000 lbs at liftoff. The penalty function is expressed by Equation 7.2, where $GLOW_{max}$ is the maximum gross liftoff weight, α is set to a value of 4 for having a steep gradient of the penalty function.

$$G = \alpha max\left(0, \frac{GLOW - GLOW_{max}}{GLOW_{max}}\right) \tag{7.2}$$

7.1.2 Step 2: Generate Concepts and Select Technologies

This problem deals with the simultaneous selection of concepts and technologies. Various number of concepts are generated, and they are summarized in Table 7.2. The concepts vary from the number of stages, the engine and propellant types, and the type of OMS and RCS. In addition to the concepts presented in Table 7.2, two continuous variables are added to the design space. These variables are the thrust-to-weight ratio and the staging velocity, as presented in Table 7.3. They are considered because their setting can influence the choice of a design. This capability shows that the new design method can also consider continuous design variables, which are discretized, during the design space exploration.

Table 7.3: Continuous variables considered during the design space exploration

Continuous variables
Thrust-to-weight ratio
Staging velocity

Technologies are also selected as means to improve the launch vehicle performance. For this problem, 10 different technologies are considered, obtained from Olds et al. [72], and Crocker *et* al. [19, 20]. The impact of those technologies on the vehicle performance and cost is presented in Step 4.

Table 7.4: Technology list with their associated system for the concept and technology selection problem.

Technology	System	Description
T1	Body	Graphite-epoxy skin construction for airframe
T2	Body	Ti/Al-SiC metal matrix composites for airframe
T3	Fuel tank	Graphite-epoxy honeycomb tanks with no liner
T4	Oxidizer tank	Al-Li 2195 stiffened skin structure
T5	Propulsion	High thrust-to-weight LOX/LH2 rocket engine
T6	Propulsion	Super lightweight LOX/LH2 rocket engine
T7	Propulsion	Electromechanical gimbal and valve actuator
T8	Propellant	Densified LH2 propellant (slush or triple-point)
T9	Electrical power	High power density fuel cells integrated with main tanks
T10	Avionics	Lightweight GN&C, RF communications, data systems, instrumentation, range safety, and controllers

7.1.3 Step 3: Populate the Morphological and Incompatibility Matrices

To facilitate the exploration of alternatives, the concepts and technologies presented in Table 7.2 are grouped into a morphological matrix. Beforehand, a set of morphological fields are generated to populate the first column of the morphological matrix. Two hierarchies of morphological fields are selected because of the complexity of launch vehicle systems. The top-level hierarchical fields are the *Overall* configuration, the strap-on *Boosters*, the *First stage* (core), and the *Second stage* (core), as displayed in the first column of Table 7.5. A series of lower-level morphological fields are then assigned to each of the top-level ones, as presented in Table 7.5. The duplicity of

clearpage

certain morphological fields is because they are applicable to more than one top-level morphological field. This is the case for the engine type, for instance, which is repeated in the *Boosters*, *First Stage*, and *Second stage* fields. This organization is used to generate and organize the concepts in the morphological matrix displayed in Table 7.6.

Table 7.5: Function and systems associated to each architectural element in the morphological matrix

Morphological fields	
Top-level	**Sub-level**
Overall	Number of stages
	Strap-on boosters
Strap-on boosters	Propellant type
	Propellant feeding type
	Thrust-to-weight ratio
	Staging velocity
First stage (core)	Propellant type
	Propellant feeding type
	OMS/RCS propellant type
	Thrust to weight ratio
Second stage (core)	Propellant type
	Propellant feeding type
	OMS/RCS propellant type
	Thrust to weight ratio
	Staging velocity
Technologies	Body
	Fuel tank
	Oxidizer tank
	Propulsion
	Electrical power
	Avionics

The morphological matrix shows the different classes of launch vehicles modeled: single-stage, two-stage with and without strap-on boosters. For each stage, a different type of propellant is available LOX/LH2, LOX/RP, and solid rocket propellants are considered, with two types of feeding approach pressure-fed and pump-fed. The first an second core stages can also use either cryogenic or storable OMS and RCS

propellants. Moreover, the stage can have different thrust-to-weight ratios, as well as different staging velocities. Finally, 10 technologies are inserted in the morphological matrix to insure simultaneous concept and technology exploration. These technologies are combined to any launch vehicle concept as an attempt to improve its performance. This morphological matrix is used for the remaining of this launch vehicle concept and technology down-selection problem.

In addition to the morphological matrix, an incompatibility matrix is generated as a means to prevent the selection of incompatible concepts on a launch vehicle architecture. Thus, a square matrix of 62 rows and columns (one for each concept) is constructed where a 0 indicates an incompatibility between two concepts and 1 indicates a compatibility. The incompatibility matrix for the concept selection problem is presented in Figure 7.1, where only the incompatibilities (0) are shown for simplicity. Although complicated at first sight, the incompatibilities in the morphological matrix usually follow general patterns. First, the concepts in a same morphological field are considered incompatible. Second, the concepts related to the overall vehicle have some incompatibilities with lower level concepts. For example, when the vehicle has no strap-on boosters or second stage, the concepts related to these two morphological fields (strap-on boosters or second stage) cannot be selected. Finally, an incompatibility is introduced when the launch vehicle has a single stage and uses solid rocket propellants because the trajectory module could not consistently converge.

7.1.4 Step 4: Map the Concepts to the Design Variables

The process of mapping the concepts to the modeling and simulation environment is the keystone element of the new concept selection process to explore a large number of alternatives. The approach, introduced in Section 5.3, is used here to map the concept and technology alternatives to the input file of RASAC-3, the sizing and synthesis code used to model launch vehicle architectures. This approach maps the

Table 7.6: Morphological matrix for the launch vehicle concept and technology exploration example

	Concepts				
Overall Vehicle					
Number of stages	1	2			
Boosters	Yes	No			
Boosters					
Propellant	LOX/LH2	LOX/RP	AP/HTPB		
Feeding	Pressure	Pump	None		
Thrust-to-weight ratio (%)	0.3	0.4	0.5	0.6	0.7
Staging velocity, ft/s	3000	4000	5000	6000	
First Stage					
Propellant	LOX/LH2	LOX/RP	AP/HTPB		
Feeding	Pressure	Pump	None		
OMS/RCS Propellant	Storable	Cryogenic			
Total thrust-to-weight ratio	1.3	1.4	1.5	1.6	
Second Stage					
Propellant	LOX/LH2	LOX/RP	AP/HTPB		
Feeding	Pressure	Pump	None		
OMS/RCS propellant	Storable	Cryogenic			
Thrust-to-weight ratio	0.65	0.75	0.85	1	
Staging velocity, ft/s	8000	10000	12000	14000	
Technologies					
Body	None	T1	T2		
Oxidizer tank	None	T3			
Fuel tank	None	T4			
Propulsion	None	T5	T6		
Hydraulics	None	T7			
Propellant	None	T8			
Electrical power	None	T9			
Avionics	None	T10			

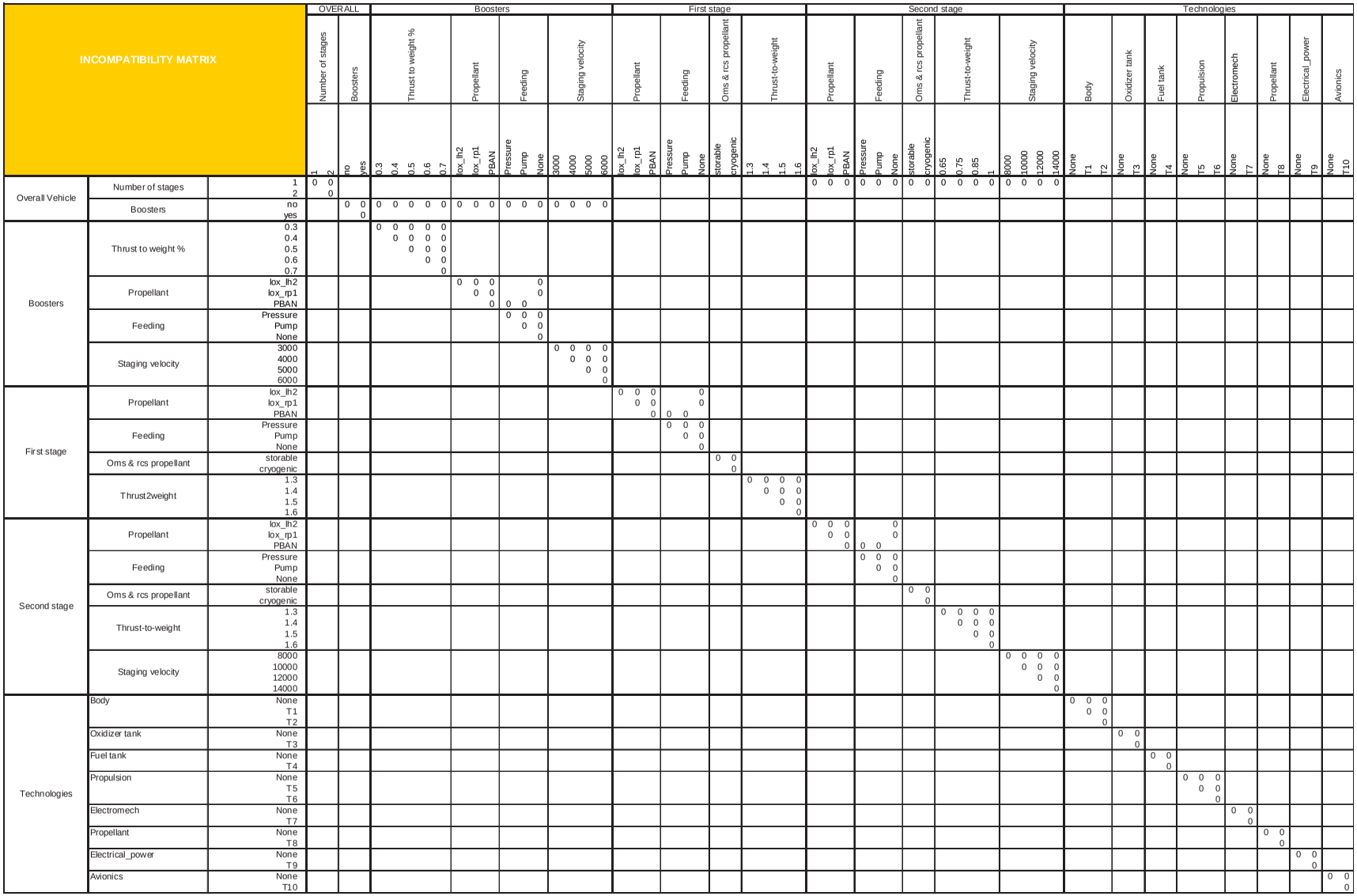


Figure 7.1: Incompatibility matrix for the concept exploration problem

concepts to different sets of input variables so that any compatible concepts can be quantitatively evaluated. Since this exercise requires several logical statements and loops, a subroutine that links the discrete design space to the continuous design space was created. This subroutine builds an input file for the six variable classes required to run RASAC-3. As a recalls, these classes: overall vehicle, mission, propulsion, geometry, mass, and cost. Once this subroutine is set up and linked to the morphological matrix, the design space can be quantitatively explored, as described in the next section.

Unlike concepts, technologies are mapped against the reduction factors (k-factors) programmed in RASAC-3, with the help of the technology impact matrix. The performance of each of the 10 technologies is computed via 13 k-factors presented in Table 7.7. Those factors assess the technology performance through the weight, propulsion, and cost are the launch vehicle disciplines. In addition, the technology impact matrix, shown in Table 7.8, is generated to model the impact of the technologies on the k-factors. For example, Technology 1 (Graphite-epoxy skin construction for airframe) is modeled by reducing the fuselage mass (K1) by 18%, and so on. Once the technologies and concepts are mapped to the modeling an simulation environment, the systems can be explored in an iterative process. The next section describes the first step of this process.

7.1.5 Step 5: Generate Compatible Architectures

The generation of compatible alternatives is done in two operations. The first one consists of selecting the concepts in the top-level morphological fields presented in Table 7.6. These fields are the number of stages, and the presence (or not) of strap-on boosters. Once these top-level morphological fields are selected, the main layout of the architecture is known, and the other morphological fields can be selected. This approach ensures that the top level concepts in the morphological matrix hierarchy be

Table 7.7: Technology reduction factors

Reduction Factor	Influence
K1	Fuselage mass
K2	Oxidizer tank mass
K3	Fuel tank mass
K4	Engine thrust-to-weight ratio
K5	Electrical conversion and distribution mass
K6	Avionics mass
K7	Engine specific impulse
K8	Liquid hydrogen density
K9	Hydraulics mass
K10	Airframe DDTE
K11	Engine DDTE
K12	Airframe TFU
K13	Engine TFU

Table 7.8: Technology impact matrix

	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11	K12	K13	K14	K15	K16
T1	0.82	1	1	1	1	1	1	1	1	1.15	1	1.15	1	0.95	1	1
T2	0.9	1	1	1	1	1	1	1	1	1.15	1	1.15	1	0.95	1	1
T3	1	0.8	0.8	1	1	1	1	1	1	1.2	1	1.25	1	1	1	1
T4	1	0.88	0.88	1	1	1	1	1	1	1.1	1	1.15	1	1	1	1
T5	1	1	1	1.12	1	1	1	1	1	1	1.05	1	1	1	1	1
T6	1	1	1	1.25	1	1	1	1	1	1	1.1	1	1	1	1	1
T7	1	1	1	1	1	1	1	1	0.85	1.05	1	1.05	1	1	1	1
T8	1	1	1	1.035	1	1	1	1.15	1	1.05	1	1.05	1	1.1	2.5	2.5
T9	1	1	1	1	0.75	1	1	1	1	1.05	1	1	1	1	1	1
T10	1	1	1	1	1	0.75	1	1	1	1.05	1	1	1	1	1	1

exploited. Otherwise, there is a stronger chance of selecting two stage vehicles. The process of selecting the vehicle concepts is done by traveling over the morphological graph, described by its adjacency matrix in Figure 7.2, where only the zeros and the upper section of the matrix are displayed for simplicity. This graph, combined to the incompatibility matrix are used to determine the possible concept combinations during the construction of a solution.

7.1.6 Step 6: Size the Architectures

The architecture sizing is performed by RASAC-3, the rapid sizing and synthesis code for launch vehicles described in Chapter 6, and using response surface equations for the trajectory analysis. The approach consists of using the input file generated in Step 5, and of running RASAC-3 until the vehicle converges on the gross liftoff weight. Once convergence of the algorithm is reached, the cost module evaluates the DDT&E and the TFU costs. The formula for the OEC presented in Equation 7.1 is then used to compute the overall vehicle performance. In the event of incompatibilities between concepts, a penalty is attributed to the value computed by the OEC. This penalty consists of multiplying the value of the OEC by a factor of 1.5, and is only necessary for the GA and SA when they generate incompatible concepts.

7.1.7 Explore the Design Space

The design space is explored using three optimization techniques: Ant Colony Optimization, Genetic Algorithm, and Simulated Annealing, which were detailed in Chapter 4. The ACO algorithm uses the settings that are shown in Table 7.9. The pheromone deposition ($\Delta\tau$) is performed according to Equation 7.3, where α is set to a value of 0.5 for the general pheromone deposition and 1 for the elitist pheromone deposition. These values seem to offer a better convergence of the ACO algorithm, along with an initial pheromone level of 1.0. Moreover, the OEC_{ref} is equal to the baseline OEC (2). The elitist ants considered for this simulation are the best ants for

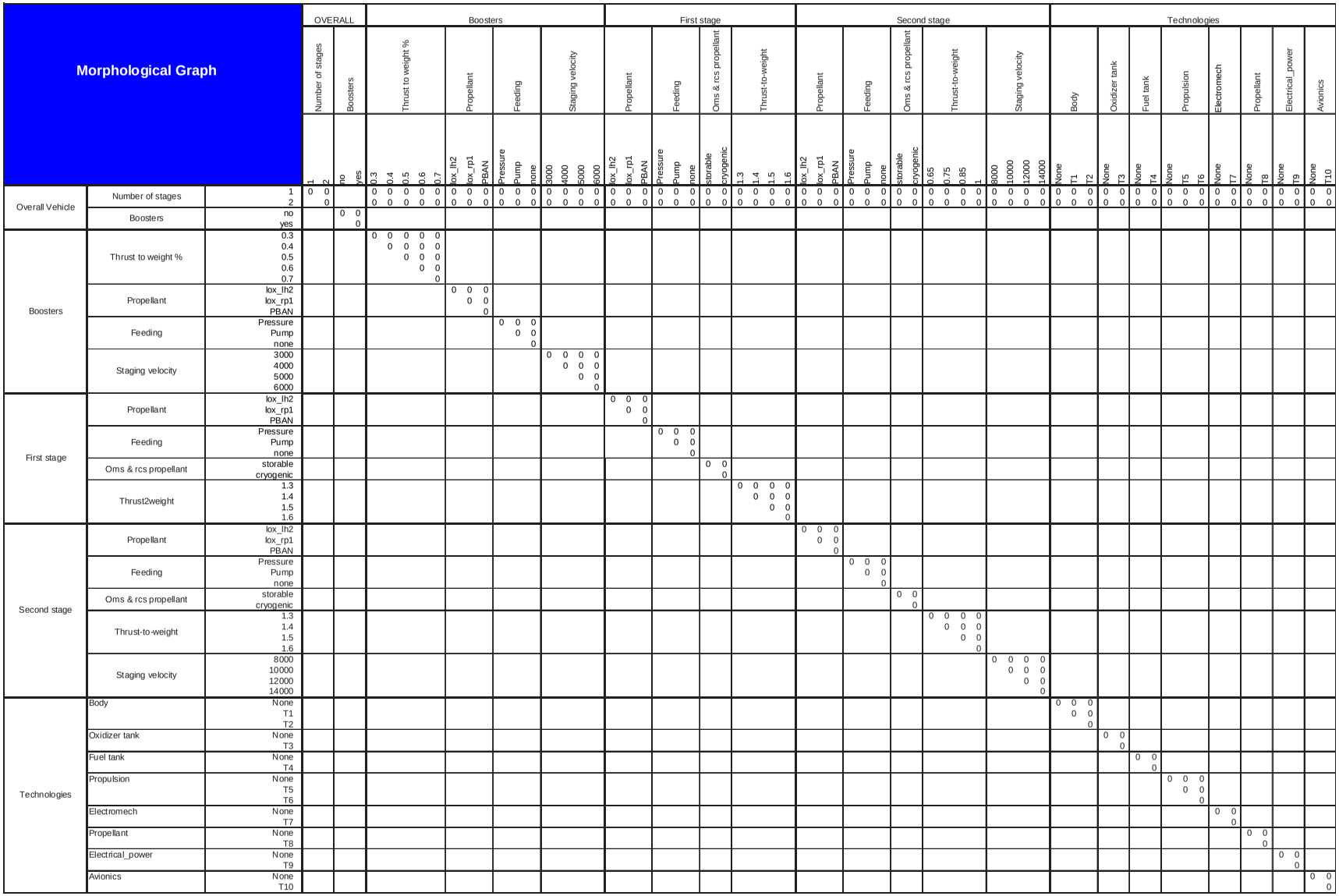


Figure 7.2: Adjacency matrix for the concept and technology exploration problem

each iteration and for the overall optimization. In other words, an additional amount of pheromone is deposited on the graph for these two elitist ants at the end of each iteration.

$$\Delta\tau = \alpha \frac{OEC_{ref}}{OEC} \tag{7.3}$$

Table 7.9: Ant Colony Optimization settings for technology selection validation problem

Optimization parameter	**Setting**
Number of ants	50
Number of iterations	30
Pheromone evaporation rate	0.25

In addition to ACO, GA is the seconde type of optimizer use to explore the design space. The chosen algorithm uses GA from the MATLAB (R2006) toolbox. The algorithm's input parameters consists of a population size of 50 with 30 generations, a crossover fraction of 0.8 and a mutation fraction of 0.01. The input parameters used for the GA are shown in Table 7.10. The algorithm is adapted, however, to take into account the hierarchy within concepts. An approach similar to Buonanno [11] using a structured Genetic Algorithm is performed. Thus, when the number of stages selected in the first morphological field was equal to 1, the bits on the GA string related to second stage options are ignored. This would insures the same likelihood of evaluating single and two-stage vehicles. The same approach is performed on the strap-on boosters.

Table 7.10: Genetic Algorithm settings for technology selection problem

Optimization parameter	Setting
Population size	50
Number of generations	30
Number of best individuals sent to next generation	2
Crossover fraction	0.8
Mutation fraction	0.01

The SA algorithm was programmed in MATLAB and use a cooling rate equal to 0.8, from an initial temperature of 600. These parameters were found to produce good convergence rate from the algorithm. The algorithm is also programmed to consider hierarchy in the string of a population member, as explained in the previous paragraph. The algorithm also has 50 initial energy states which are cooled 30 times. Table 7.11 summarizes the SA settings.

Table 7.11: Simulated Annealing settings for technology selection problem

Optimization parameter	Setting
Initial temperature	600
Cooling rate	0.8
Number of initial states	50
Number of iterations	30

7.2 Analysis of the Results

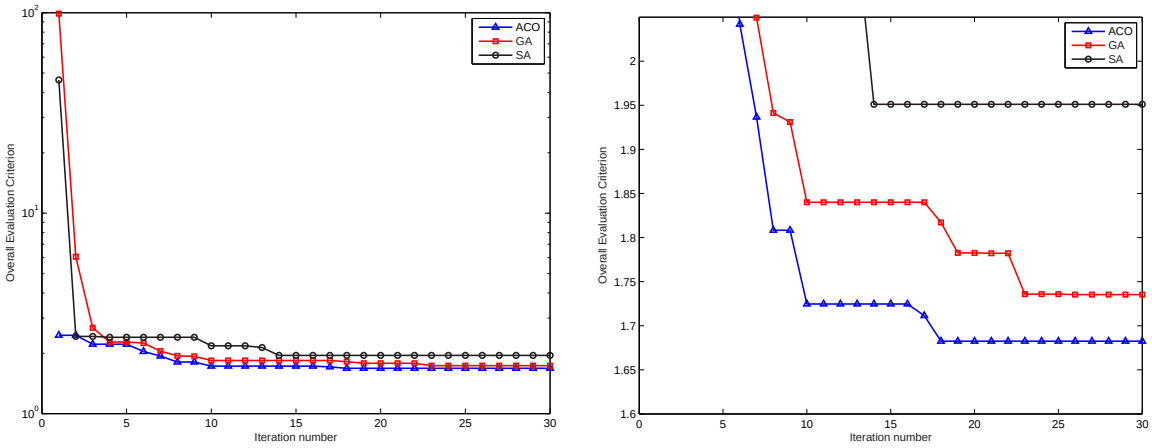
The design space was explored as a result of the application of the six-step concept selection process described in the previous sections. The next paragraphs show the results obtained from this exercise. Three types of analyses are generated: 1) comparison of stochastic optimization approaches, 2) analysis of the launch vehicle selected, and 3) analysis of pheromone matrix. The latter is performed because

7.2.1 Optimizers Comparison

Similar to the technology selection problem, the convergence history and algorithm error is compared between the three stochastic optimization algorithms (ACO, GA, and SA). A total of 124 simulations are performed, where each simulation requires 1500 function calls from RASAC-3. The computation time per function call is approximately 0.5 second on a 2.8 GHz Intel Pentium D micro-processor. Figures 7.3 shows the evolution of the objective function for the ACO, GA, and SA as a function of the number of iterations for the first simulation. More precisely, Figure 7.3a shows

the OEC variation on a logarithmic scale, and Figure 7.3b shows a zoom of the evolution of the algorithm on a linear scale. The curves show that ACO finds a better objective function with an OEC equal to 1.68, than GA and SA, with OECs equal to 1.73 and 1.95, respectively.

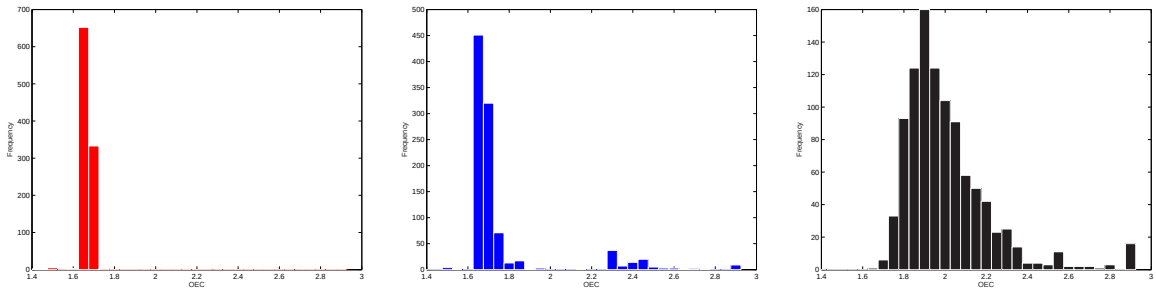
Figure 7.3 also shows a better performance of the ACO algorithm during the first iterations. This enhanced performance is caused by the consideration of concept incompatibilities during the solution construction phase of ACO. The other two algorithms, GA or SA, do not account for incompatibilities during the construction of solutions, which results in the generation of poor concepts at the beginning of the optimization process. The two algorithms (GA and SA) thus spend time considering incompatible solutions, which can be observed by the large OEC value in the case of GA and SA (Figure 7.3).



(a) Log (b) Zoom

Figure 7.3: Variation of the overall and iteration best solution as a function of the iteration number

Since ACO, GA, and SA are stochastic optimization algorithms, the performance of the algorithm is evaluated statistically. A total of 1,000 simulations are performed for each of the three stochastic algorithms. Their distribution plots are displayed in Figure 7.4, which shows that ACO consistently outperform the GA and SA over the 1,000 simulations. This can also be observed by computing the average and standard



(a) ACO (b) GA (c) SA

Figure 7.4: Distribution of the objective function for the 124 cases

Table 7.12: Comparison between the performance of ACO, GA, and SA for the concept selection problem

	ACO	GA	SA
Average	1.6706	2.1703	2.0338
Standard deviation	0.0248	3.1949	0.4129

deviation over the 1,000 simulation, as shown in Table 7.12. The table shows that the ACO mean is lower than the other two stochastic algorithms, and it also has a lower standard deviation. A t-test is performed on comparing the ACO results to GA and SA, and the test results in the rejection of the null hypothesis with a significance level of 0.005. That is, the ACO mean is lower than the GA's and SA's with a confidence of 99.5%, meaning that ACO converged to a cheaper launch vehicle concept.

7.2.2 Analysis of the Best Concepts

An analysis of the results generated in the previous section shows that 11 of the top 15 concepts generated by the ACO simulation are two-stage vehicles with LOX/LH2 pump-fed engines on both stages. The remaining are two-stage vehicles with LOX/RP1 pump-fed engines on the core first stage and LOX/LH2 pump-fed engines on the second stage.

Tables 7.13 and 7.14 show a ranking of the optimal 15 solutions obtained from an ACO simulation, with their respective concepts. This visualization technique could help design engineers to associate the physical arrangement of optimal concepts, and to analyze the difference between them. This is presented in the next paragraph. The two tables show the cost and mass breakdown for each stage, as well as a description of the concept. One can see that the two main concepts selected are two stage launch vehicles where the core first stage is powered by pump-fed LOX/LH2 engine core second stage, and the core second stage is powered by pump-fed engines with either LOX/RP or LOX/LH2 propellants. In terms of the weight breakdown one can see that LOX/RP concepts lead to heavier launch vehicles because of a larger quantity of propellant required, and a lower engine specific impulse. The dry mass fraction of LOX/RP first stage, however, is lower than the LOX/LH2 second stage because of a lower propellant volume.

Table 7.13: Best concepts description summary

Rank	OEC	Stage 1					Stage 2				
		DDTE $M	TFU $M	GLOW lbm	Dry mass lbm	Propellant lbm	DDTE $M	TFU $M	Gross mass lbm	Dry mass lbm	Propellant lbm
1	1.6829	8,272	2,126	6,040,784	378,671	4,396,006	3,528	797	1,103,291	81,925	866,564
2	1.6831	8,216	2,129	6,033,694	378,661	4,391,121	3,522	805	1,101,554	82,155	865,220
3	1.6982	8,569	2,246	6,457,320	407,109	4,949,786	3,228	721	915,656	70,645	693,767
4	1.7386	8,695	2,387	4,655,437	409,468	3,188,431	3,046	708	938,024	66,073	719,860
5	1.7469	8,567	2,280	4,516,552	397,983	2,888,614	3,457	792	1,123,070	79,409	888,698
6	1.7657	8,234	2,154	6,024,174	381,029	4,094,779	3,967	943	1,388,726	101,987	1,127,824
7	1.7668	8,773	2,302	6,631,789	434,454	4,824,853	3,406	802	1,193,557	80,078	956,945
8	1.7689	8,470	2,307	4,543,124	365,740	2,936,449	3,549	829	1,129,887	79,311	895,427
9	1.7775	8,587	2,337	4,693,934	415,489	3,019,445	3,493	815	1,147,959	83,474	909,265
10	1.7785	8,612	2,329	4,626,533	399,153	2,974,979	3,512	818	1,141,606	82,262	904,198
11	1.7856	8,635	2,343	4,559,391	421,475	2,918,124	3,546	815	1,114,373	85,534	874,033
12	1.7859	8,799	2,460	4,790,588	424,946	3,304,134	3,150	737	946,498	70,638	724,293
13	1.7873	8,962	2,355	6,376,137	442,209	4,595,857	3,350	786	1,168,838	76,971	935,611
14	1.7877	8,779	2,314	6,121,183	399,129	4,433,082	3,542	827	1,124,689	79,017	890,619

Table 7.14: Best concepts weight breakdown

Rank	Stage 1				Stage 2			Technologies
	Engine type	RCS	T/W	Vs. ft/s	Engine type	RCS & OMS	T/W	
1	lox_rp1_pump	storable	1.3	12,000	lox_lh2_pump	storable	1.00	T6, T9, T10
2	lox_rp1_pump	storable	1.3	12,000	lox_lh2_pump	cryogenic	1.00	T6, T9
3	lox_rp1_pump	storable	1.3	14,000	lox_lh2_pump	storable	1.00	T9, T9
4	lox_lh2_pump	storable	1.4	14,000	lox_lh2_pump	storable	0.65	T6
5	lox_lh2_pump	storable	1.5	12,000	lox_lh2_pump	cryogenic	0.85	T6, T9, T10
6	lox_rp1_pump	storable	1.3	10,000	lox_lh2_pump	cryogenic	1.00	T6
7	lox_rp1_pump	storable	1.3	12,000	lox_lh2_pump	cryogenic	0.65	T5, T9, T10
8	lox_lh2_pump	cryogenic	1.3	12,000	lox_lh2_pump	cryogenic	0.85	T6, T8, T9
9	lox_lh2_pump	storable	1.4	12,000	lox_lh2_pump	cryogenic	0.85	T5, T9
10	lox_lh2_pump	storable	1.4	12,000	lox_lh2_pump	cryogenic	0.85	T6, T10
11	lox_lh2_pump	storable	1.5	12,000	lox_lh2_pump	storable	1.00	T5, T9
12	lox_lh2_pump	cryogenic	1.3	14,000	lox_lh2_pump	cryogenic	0.75	T5
13	lox_rp1_pump	cryogenic	1.5	12,000	lox_lh2_pump	cryogenic	0.65	T6, T9
14	lox_rp1_pump	cryogenic	1.4	12,000	lox_lh2_pump	cryogenic	0.85	T6, T8, T9

A comparison between the first stage concepts of Table 7.13 shows that a LOX/RP first stage (Rank #3) can be cheaper and can have a lower mass fraction than a LOX/LH2 stage (Rank #1 and #2). However, because of the lower staging velocity, the second stage of the concept ranked #3 is heavier, has a larger dry mass fraction, and is more expensive. This trend thus leads to a more expensive LOX/RP first stage launch vehicle and a larger OEC, compared to the LOX/LH2 first stage concepts (Rank #1 and #2). Moreover, the large thrust-to-weight ratios shown in Table 7.14 are explained by the selection of the RSE boundaries. It was experienced that low thrust-to-weight ratios (e.g., 1.2), combined to low booster staging velocities, for instance, could not generate feasible launch vehicle trajectories. For this reason, the lower bound for the liftoff thrust-to-weight ratio in the trajectory DOE for the was increased to 1.3, insuring the generation of feasible trajectory solutions. This is why the thrust-to-weight ratios displayed in Table 7.14 are larger than expected.

Taking the best concept from Table 7.14, one can analyze the concept in more detail. First, Table 7.15 shows a description of the concept by showing the attributes selected from the morphological matrix. The launch vehicle architecture is a two-stage vehicle, with LOX/RP and LOX/LH2 pump-fed liquid rocket engines on the first and second stage, respectively. The core first stage has a thrust-to-weight ratio of 1.30 and stages at a relative velocity of 12,000 ft/s, and the second stage has a

thrust-to-weight ratio of 1.0. The OMS and RCS use storable propellants. Then, the mass is then broken down in Tables C.4 and C.5 for the first and second stage masses, respectively. The vehicle has a GLOW of 6,040,784 lbm, and the second stage has gross mass of 1,103,342 lbm. The dry mass of the first and second are equal to 378,671 lbm and 81,925 lbm, respectively.

Table 7.15: Best concept selected from the morphological matrix

Morphological fields		
Level-I	**Level-II**	**Concept selected**
Overall	Number of stages	2
	Strap-on boosters	No
First Stage	Thrust-to-weight ratio	1.30
	Engine type	LOX/RP, pump-fed
	RCS propellant type	Storable
	Staging velocity, ft/s	12,000
Second Stage	Thrust-to-weight ratio	1.00
	Engine type	LOX/LH2, pump-fed
	OMS and RCS propellant	Storable
Technologies	Body	None
	Tanks	None
	Propulsion	T6
	Propellant	None
	Electrical	T9
	Avionics	T10

Finally, the cost of the two vehicles is displayed in Tables 7.16 and 7.17, The development cost is $8,271.6 million and $3528.4 million for the first and second stage, respectively; and the theoretical first unit cost is $2,126.2 million and $796.8 million, respectively.

This analysis of the best concepts obtained from the ACO algorithm shows how the proposed design methodology can generate traditional mass and cost breakdown results, in addition to the analysis of the OEC. The methodology is thus useful in exploring vehicle concepts, and the best concepts can be analyzed further by design engineers.

Table 7.16: First stage cost breakdown

Group	DDTE, $M (2002)	TFU, $M (2002)
1.0 Body	119.28	627.32
2.0 TPS/TMS	414.11	43.82
3.0 Main propulsion	2368.44	298.76
4.0 RCS/OMS	286.21	336.43
5.0 Primary power	77.31	60.69
6.0 Electrical	257.77	144.40
7.0 Hydraulic	166.18	33.61
8.0 Avionics	96.26	6.77
9.0 Processing	3606.33	280.30
Margin	879.70	294.11
Total	8271.6	2126.2

Table 7.17: Second stage cost breakdown

Group	DDTE, $M (2002)	TFU, $M (2002)
1.0 Body	58.47	253.34
2.0 TPS/TMS	219.12	19.45
3.0 Main propulsion	1089.43	104.88
4.0 RCS/OMS	135.53	119.28
5.0 Primary power	33.11	22.77
6.0 Electrical	97.50	47.02
7.0 Hydraulic	54.66	8.12
8.0 Avionics	67.58	4.68
9.0 Processing	1414.42	105.62
Margin	358.58	111.63
Total	3528.4	796.8

7.2.3 Analysis of the Pheromone Matrix

The third type of analysis of the results consists of looking at the distribution of pheromone on the pheromone matrix. As a reminder, the pheromone matrix is a way to identify combination of concepts that are likely to lead to good solutions. The matrix can also be seen as a histogram, or a contour plot showing the goodness of combinations between two concepts. The pheromone matrix that is created at the end of the first ACO simulation is shown in Figure 7.5, where the different colors represent the variation of the pheromone value on each edge of the morphological graph. A red color represents high pheromone concentration, whereas a purple one represents a low pheromone concentration. Moreover, the cells in the matrix related to the the number of stages and the boosters are removed, for simplicity, and only the upper portion of the matrix is shown because of its symmetry.

The pheromone matrix is an interesting way to look at good concept combinations. In short, good solutions tend to be generated more frequently by depositing a higher concentration of pheromone on the path used to generate the solution. After the iteration reaches convergence, high pheromone concentrations can be observed, as depicted in Figure 7.5. For example, it shows that first and second stages pump-fed engines are more likely to generate good solutions, as seen by the high pheromone level on the appropriate cell in the pheromone matrix.

7.3 Findings

The results from the previous section first illustrate that modeling the concept selection problem by a graph can facilitate the concept exploration process by programming the architecture hierarchy into the design process. The combination of the adjacency matrix and the morphological matrix is efficient in ensuring compatible concepts and constructing hierarchical solutions, such as when one wants to explore the options of morphological matrix. This approach eliminates the need of using

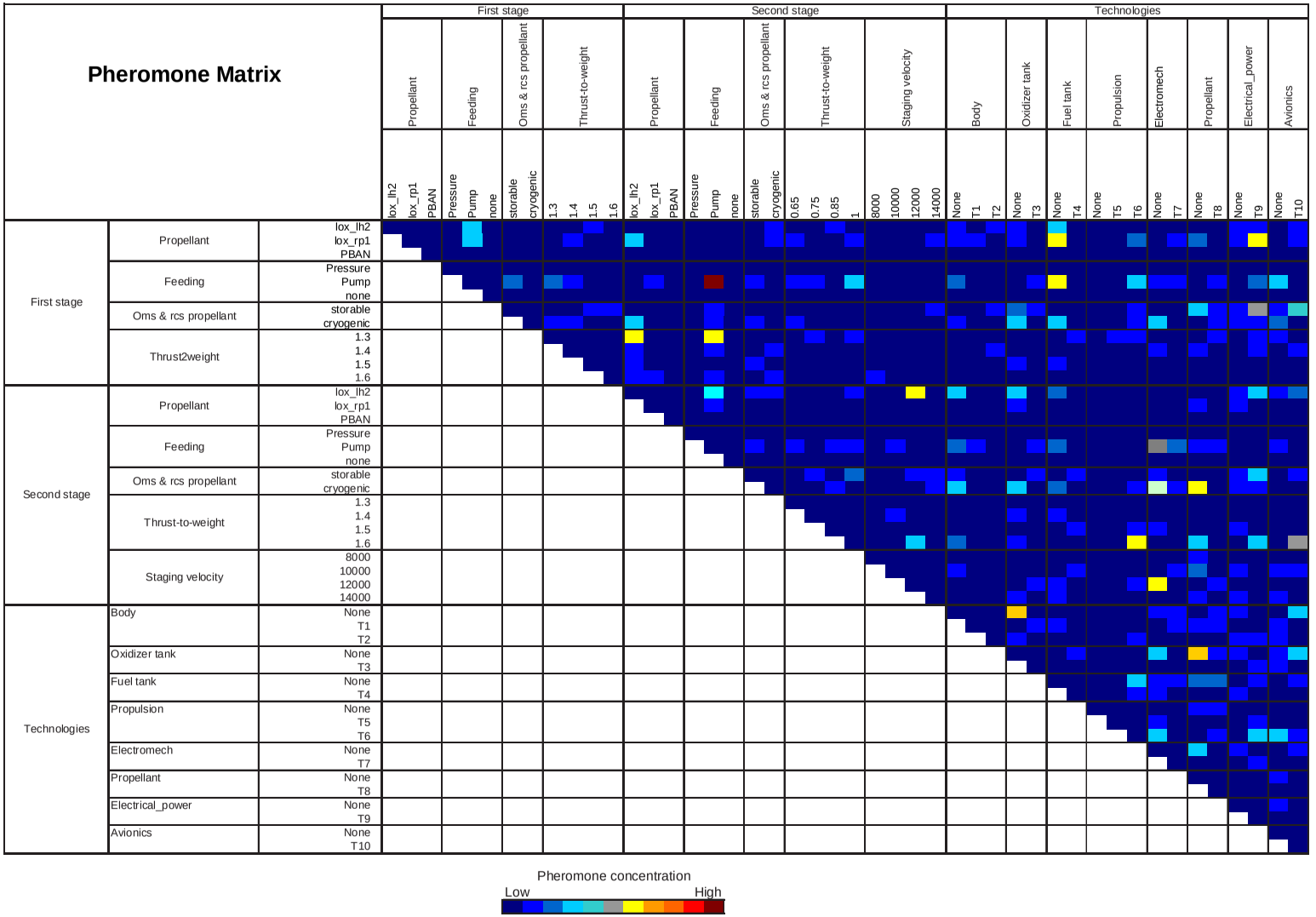


Figure 7.5: Contour plot of the pheromone matrix

penalty functions, and it helps to generate compatible alternatives that can be easily analyzed in the modeling and simulation environment.

Furthermore, the application of ACO, GA, and SA to explore the design alternatives is another interesting finding. The better performance of ACO is once again due to the hierarchical solution construction method and the avoidance of generating incompatible solutions. Similar to the technology selection problem, an evaluation of the optimization algorithms is done. This evaluation shows that ACO found more optimal solutions than GA and SA for a launch vehicle concept and technology selection problem.

As stated in the introductory paragraph, the objective of this chapter was to demonstrate the concept and technology down-selection methodology within a launch vehicle architecture study. The results presented in the last section show how the methodology can explore the design space for the design engineers to help their decision making process during architecture selection. Appendix D presents another application of the methodology on an aircraft technology selection problem. The next chapter discusses the results and show how they can be interpreted to address the research questions and hypotheses developed for this thesis.

CHAPTER VIII

DISCUSSION

Chapter 7 showed the implementation of the proposed concept and technology down-selection methodology on a launch vehicle architecture. This chapter revisits the research questions and hypotheses developed in Chapter 3 (Section 8.1), and compares the proposed methodology with existing concept and technology selection approaches (Section 8.2).

8.1 Revisiting the Research Questions and Hypotheses

The goal of this section is to revisit the research questions and hypotheses and use the results generated in the previous chapter to address them. They constitute the backbone of the research performed in this thesis and their answering is necessary in closing the discussion of this thesis.

To help the discussion, three design approaches are compared to address the research questions and hypotheses. They are enumerated below:

Approach 1: Man-in-the-loop exploration

Approach 2: Standard discrete optimization algorithm (GA, SA)

Approach 3: Graph-based design space exploration using ACO

The first approach is the traditional design process where design engineers pick a certain number of alternatives from the morphological matrix, which are then evaluated quantitatively. The approach selects the architecture based on a manual design

space exploration. The second approach consists of modeling the concepts in a discrete cartesian design space and uses a genetic algorithm or simulated annealing to perform the exploration of the design space. A mix of continuous and discrete variables is considered, and the best architecture is the one that is the most Pareto optimal. Finally, the third approach is the new architecture selection process that models the morphological matrix by a graph and then uses the incompatibility matrix to generate compatible solutions. The approach is described in Chapter 3. These three approaches are thus assumed to represent well the different possibilities in concept selection, discussed in Chapter 2. The next sections compare them to answer the first research question.

8.1.1 Revisiting Research Question 1 & Hypothesis 1

The first research question is a response to the main problem of this thesis, that is, enabling the quantitative exploration of concepts early in the conceptual design phase. It refers to the use of a design approach that broadens the design space during architecture selection.

Research Question 1: How can the design space of an aerospace vehicle be explored more broadly during its concept selection?

To answer this question, the reader may recall the results of Chapter 7, which show that the proposed design methodology enables the design space exploration launch vehicle architecture with technologies. These results are used to address the first research question, by comparing each method between each other, as shown in the next paragraphs.

First, what is the performance Approach 1 relative to Approaches 2 and 3? The results presented in the launch vehicle concept selection problem show that Approach 3 explores more broadly the design space. The design engineers themselves cannot

explore as efficiently a design space as a discrete optimization algorithm. Combinatorial optimization algorithms such as ACO are more capable of exploring large discrete design spaces than humans because of their search efficiency and their capability of quickly generating a large number of architectures. Moreover, in optimizing a morphological matrix with technologies, ACO performs better than GA and SA by finding consistently better solutions. Therefore, the answer of the first research question resides in the use of a graph-based optimization approach for exploring hierarchical systems. The discussion can now be turned toward the first hypothesis, which is done in the next section.

The elaboration of the first hypothesis is directly related to the first research question. It supposes that a graph could facilitate and accelerate the construction of compatible architectures. It is presented below as an aid to the reader.

Hypothesis 1: Modeling the morphological matrix by a graph accelerates and facilitates the construction of physically compatible solutions.

Answering the first hypothesis is done with the similar logic as the first research question since they are closely related. However, the fundamental difference is that to prove hypothesis 1, one may ask which of the three above design approaches generate faster and more easily compatible solutions. During the implementation of the launch vehicle concept selection problem in Chapter 7, it was shown that the application of GA or SA requires the use of penalty function to account for concept incompatibilities. These two algorithms thus produce incompatible solutions, which translate into a less efficient design space exploration. Therefore, the use of a graph to generate compatible solution can speed up and help the generation of compatible solutions in the same amount of time. Moreover, other vehicle design morphological matrices can have significantly more incompatibilities, which make the use of GA and SA difficult. The method proposed in this thesis could then be even more beneficial in those circumstances.

8.1.2 Revisiting Research Question 2 and Hypothesis 2

Now that the research question and hypothesis relative to the solution construction are addressed, the second research questions, enumerated below, concerns the potential use of an optimization algorithm to enable the exploration over a large combinatorial design space.

Research Question 2: What design exploration approach could reduce the exploration time to reach the optimality of the solution when exploring aerospace system architectures?

Since it is concluded that a graph can help the generation of compatible solutions, this research question leads also to the hypothesis that a graph-based optimization approach, named Ant Colony Optimization, would reduce the exploration time and the optimality of the solution for the exploration of the combinatorial design space.

Hypothesis 2: Ant Colony Optimization (ACO) can find more optimal solutions when exploring aerospace system architectures.

The second hypothesis was elaborated because of the promising performance of Ant Colony Optimization on the exploration of problems modeled by graphs. The application of the technology and concept down-selection methodology on a launch vehicle architecture (Chapter 7) showed that ACO finds generally more optimal solutions that GA and SA. This shows that the ACO algorithm is well tailored to performing a quantitative concept and technology exploration in presence of concept incompatibilities, and hierarchy. It thus confirms the validity of the second hypothesis.

8.1.3 Revisiting Research Question 3

Research Question 3: What optimization approach enables a better simultaneous concept and technology exploration of aerospace architectures ?

As a recall, the launch vehicle concept selection problem showed how ACO outperformed GA and SA in simultaneously exploring technologies and concepts. These results showed that for exploring a total of 44 concepts and 10 technologies, ACO found continuously more optimal solutions than GA and SA. Therefore, it can be said that ACO, because of its capability of constructing solutions with consideration of incompatibilities and the hierarchy within the concepts of a system, can find more optimal solutions in simultaneously exploring a concept and technology design space.

8.2 Benchmarking

Before going into the benchmarking of the method relatively to other methods, it is worth determining if the results generated in the previous chapter confirmed that the design process was rightly designed. The technology selection problem (Appendix D) generated a large amount of data to answer this question. Here are two arguments justifying its verification. The first argument consists of determining if the proposed process performs the intended functions. The implementation the six steps of the design process for the problems addressed in Chapter 7 and Appendix D show that the design process indeed perform its intended functions. It formulates the problem (Step 1), find and organize alternatives (Steps 2 and 3), map them to the modeling and simulation environment (Step 4), generate feasible architectures (Step 5), evaluate the architectures (Step 6), and to quantitatively explore them. The results in each of the steps presented in Section D.1 prove that the design process perform its intended functions. The second argument is that it produces similar results when compared to a traditional alternative selection approach widely accepted in aerospace engineering. The process is accurate and repeatable, which are two important criteria for verification purposes. For these two reasons, the technology selection process verifies the concept and technology selection process was rightly developed.

Then, the following thesis objectives were developed as a way to benchmark the

various design methods. These objectives were translated in a set of four requirements, which were elaborated in Chapter 2 to evaluate design processes. These requirements are repeated in Table 8.1 for simplicity .

Objective 1: Expand the size of the design space explored during the concept selection of aerospace architectures

Objective 2: Improve the means for determining the optimal architecture and to reduce the computation time to reach this optimality.

Objective 3: Enable the selection of the concepts and technologies simultaneously and earlier in the conceptual design process.

Table 8.1: Design process evaluation criteria

Criteria	Description
1	Execution time
2	Convergence error
3	Capability to quantitatively explore a morphological matrix with incompatibilities
4	Adaptability to new concepts and technologies

These criteria are used to evaluate the new concept and technology exploration process, and compare it to the other concept and technology exploration processes. This evaluation is presented in Table 8.2, where each criterion is referred to the list in Table 8.1, and where a letters from A to E represent a good to a poor performance, respectively.

First, one can see that the execution time and convergence error criteria are both assigned values of B. The architecture selection problem in Chapter 7 have shown that graph-based optimization algorithms converge relatively rapidly and close enough to the best known value of the design space. For this reason, the new design process is considered as *good*, compared to the other design processes and is assigned a letter B. Furthermore, the new design process really outperforms the other design processes

with its capability of exploring technologies and concepts simultaneously (Criterion 3). The performance of this criterion was shown by the successful implementation for architecture selection where technologies are simultaneously explored with the concepts. Finally, the capability of the design process to easily integrate new concepts and technologies in the exploration of alternatives (Criterion 4) explains the A in the last column of Table 8.2. This adaptability was shown in Chapter 5 where the text-based morphological matrix can easily be updated and create a new concept exploration problem.

Table 8.2: Benchmarking of the concept selection methods

Method	Author(s), year	Criteria			
		1	2	3	4
sGA	Buonanno, 2005 [11]	B	B	C	B
Subsystem selection	Mattson & Messac, 2002 [63]	A	C	B	A
IRMA	ASDL, 2006 [64]	E	E	D	B
Decision Matrix	Pugh, 1996 [76]	E	E	D	A
TIES\GA	Kirby, 2004	A	C	C	C
Thesis	*Villeneuve, 2007*	B	B	A	A

CHAPTER IX

CONCLUSION

9.1 Summary

The focus of this thesis was on the development of a design methodology that enables the concept and technology exploration of complex aerospace architectures. This is a response to a growing need of designing large system architecture with a large number of systems.

The problem statement at the source of this thesis is that the concept and technology selection process occurring during the early phases of conceptual design has to explore a discrete design space that grows factorially with the number of concepts. In relation to this complexity, the solution space is composes of concept incompatibilities, which make the design space exploration more difficult.

To address this problem, a six-step design method is created. This method, presented in Chapter 3 enables the simultaneous design space exploration of concepts and technologies by modeling the design space with a graph and linking an incompatibility matrix to it. In addition, the application of graph-based combinatorial optimization approach enabled the exploration of a discrete design space more efficiently than traditional man-in-the-loop approaches.

The new design methodology has two major applications. First, it enables the quantitative design space exploration of a morphological matrix populated with incompatible concepts. This contribution opens up the way to a more thorough design space exploration. Second, it enables the simultaneous exploration of concepts and technologies. This thus ensures a better selection of both entities in an environment where they are highly coupled.

9.2 Summary of Contributions

The first contribution of this thesis regards the development of a formal design process to quantitatively and simultaneously explore concept and technology alternatives of large aerospace architectures. This design process was applied to two concept exploration problems: technology selection, and simultaneous concept and technology selection.

As a second contribution, the modeling of the design space by a graph, as well as the development of a graph based solution construction approach. The graph and the solution construction enables a novel approach of constructing hierarchical architectures and exploring a design space populated by incompatible concepts. This resolves one of the difficulties in quantitatively exploring a morphological matrix.

The application of a graph-based optimization algorithm such as ACO is the third contribution of this thesis to the knowledge base. ACO is used to explore technology sets for the first time, and was found useful in visualizing the design space, using the pheromone matrix generated. The third contribution can be summarized by the quantitative optimization of launch vehicle architectures.

Finally, the fourth contributions resides in the development of a modeling and simulation environment for launch vehicles, named the Rapid Access-to-Space Analysis Code-3 (RASAC-3). The program is capable of modeling a large variety of launch vehicle concepts. It sizes the vehicle with rapidity and can be easily modified to integrate new design concepts in the exploration process if they are not modeled by the existing disciplinary modules.

9.3 Pitfalls

Some pitfalls should also be addressed at this point following the demonstration of this alternative concept and technology down-selection methodology. First, the design requirements can significantly change the outcome of the algorithm. For example,

changing the figures of merit in the overall evaluation criterion can significantly change the concepts and technologies in the optimum concepts. It is thus recommended to link the proposed methodology with a requirements definition process to guarantee their proper selection.

Second, stochastic optimization algorithms such as ACO can get trapped into local minima, or to near-optimum solutions. Indeed, a discrete optimization algorithm cannot guarantee finding an absolute minimum when the design space has a large number of combinations. It is thus recommended that design engineers optimize the design space with repetitions, to insure a better solution from the methodology. Moreover, because they can converge to near-optimal solutions, design engineers should perform a localized optimization of the response output by the algorithm.

Finally, the goal of the optimization algorithm and of the proposed methodology is to help design engineers in exploring a larger combinatorial design space, and finding new solutions within this design space. This help should therefore facilitate their decision-making process during the concept and technology down-selection phase occurring in conceptual design.

9.4 Recommendations for Further Research

Enhancement of the Optimization Algorithm

As discussed by Dorigo [28], linking the combinatorial optimization algorithm to a local search approach usually leads to better solutions and faster convergence. It would thus be interesting to explore this avenue by programming a local search optimizer inside the ACO algorithm to see if it improved the algorithm performance. Moreover, the addition of a tabu list [36] (i.e., a list that avoids of repetition of the same solution during the iterations) to the ACO algorithm to ensure that the design space is explored more thoroughly.

In addition to the local search algorithm, looking developing an exact optimization method be more efficient in exploring the design space than a meta-heuristic algorithm such as ACO. It is known that exact methods (i.e., optimization algorithms that are problem-specific) have usually a better performance than meta-heuristic algorithms[31]. However, there exist no such method at the moment for concept exploration. The development of an exact method for this type of problem could potentially reduce the convergence error and rate.

Improve the accuracy of the Modeling and Simulation Environment

Improving the modeling and simulation environment with more accurate disciplinary models is another future area of investigation. Higher fidelity models are available in the aerospace industry, which could enhance the accuracy of the launch vehicle concept modeled. These models could be integrated into a different sizing and synthesis code for launch vehicles, and could generate a better confidence in the actual responses generated from the application of the methodology. Higher fidelity models, however, can introduce more complexity and less flexibility into the modeling and simulation environment. The design engineers must therefore be aware of this tradeoff, and choose the disciplinary models by taking these considerations into account.

Demonstrate the Methodology on a Different Aerospace Architectures

In addition to the optimization, it would be interesting to demonstrate the methodology on different aerospace architectures. The High Altitude Long Endurance Unmanned Aerial Vehicle [71] is an example where the methodology could be demonstrated because of the large number of concepts and technologies related to this system. The launch vehicle concept and technology exploration problem is generic since it covers a large variety of concepts but implementing the method on another

system would improve the confidence in the process.

Application to System of Systems

Systems engineering has recently seen the emergence of system of systems engineering. Although slightly different than the concept and technology selection problem, the new design process could potentially be used to explore the alternatives arising in systems of systems problems. Actually, the method does not differentiate if the discrete choice is a system, a concept or an alternative. For this reason it is believed that this design method could be applied to this emerging field.

APPENDIX A

REVIEW OF THE SPACE SHUTTLE DESIGN PROCESS

A.0.1 The Early 60's

Early studies of low cost space vehicle started in the early 1960's. As the race for the Moon was still ongoing, NASA's engineers were setting forth the next activities that would continue to stimulate the aerospace industry after the Apollo era. Prior to 1960, individual concept studies were completed on general vehicle architectures without any clear design requirements. During the early 60's, however, NASA established its future goals. These goals could be summarized by the U.S. future needs of a launch vehicle capable of flying frequently and economically to low earth orbit (LEO). NASA wanted an affordable vehicle capable of frequently transporting heavy payloads to low earth orbit, in order to assemble a space station and build on space flight experience for Mars explorations [45]. This would denote the beginning of the architecture selection and conceptual design, which lasted until 1972.

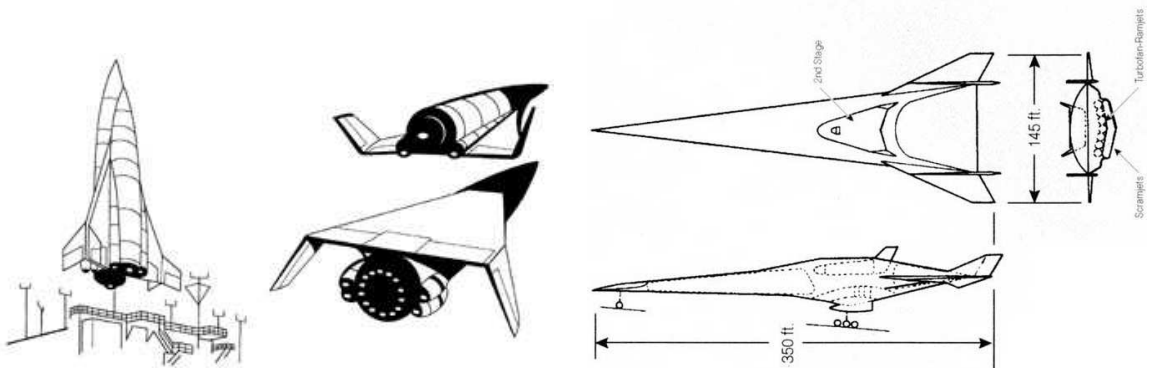
With this new set of requirements, approved by NASA engineers, research for various launch vehicle architectures began. The focus was directed toward reusable launch vehicles. It was indeed believed that only reusable vehicles would be capable of satisfying the aforementioned goals. With only expendable vehicles to rely on, it became a difficult task since no historical data really existed on this type of vehicle. This brought many difficulties to the program but made the Space Shuttle one of the most innovative aerospace vehicles of its time. Nevertheless, three major architectures attracted more consideration in the beginning of the design process [45]: the

Aerospaceplane, the Nexus, and the Astrorocket (see Figure A.1).

The Aerospaceplane was one of the promising concepts. Research on the Liquid Air Cycle Engine as well scramjets engines were believed to be promising, and the NASA wanted to bring launch vehicle operation turnaround to a level similar to the airline industry in order to accommodate those flights per year and cost per flight objectives. The potential reduction in vehicle weight due to a much lighter load in oxygen requirement was also in favor of these two concepts. Liquid Air Cycle Engine was in fact going to be an efficient way of filling the vehicle with liquid oxygen as it flies in the atmosphere, and then use it as oxidizer for the rocket operation. Research at Marquardt Corp., a Los Angeles propulsion research company, showed the technical feasibility of this concept [45]. Airbreathing propulsion also seemed to be promising and benefiting from a specific impulse standpoint. The technical challenges of supersonic combustion stability, and the problem of producing positive thrust out of a scramjet contributed to reservations of both technologies from the aerospace community.

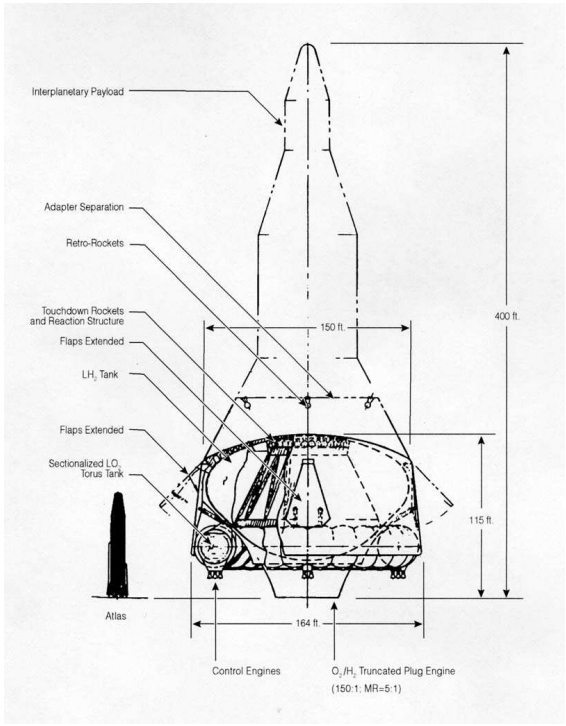
The Nexus was a large space vehicle. Its purpose was to to fulfill NASA's heavy-lift requirements for accommodating bigger payloads. This philosophy was believed to bring down the cost of a assembling a space station; given that fewer launches had to be done. The vehicle was designed to takeoff and land horizontally, eliminating the need for wings and high-heat reentries. It was also designed for a payload capacity of 2,250,000 lbs to low earth orbit and a takeoff gross weight of 48,100,000 lbs [102].

The Astrorocket was designed to deliver a lighter payload than the Nexus: 37,000 lbs to LEO. It was designed to fly 240 times per year and cost $1.5 M per flight [45]. This Two-Stages-to-Orbit (TSTO) vehicle had a liftoff weight of 2,800,000 lbs. Both the first and the second stages had lifting body shapes and liquid hydrogen (LH2) and liquid oxygen (LOX) rocket propulsion engines. The design mainly relied on the Dyna-Soar (X-20A)studies, a military spacecraft, for the selection of the vehicles



(a) Astrorocket

(b) Aerospaceplane



(c) Nexus

Figure A.1: Major space shuttle concepts by the end of 1972. Source: Heppenheimer, 2002 [45]

shape and the thermal protection system.

Although these concepts could not get approval inside NASA, they stimulated creativity, which yielded to more design studies that were completed a few years later. Designers were still focusing more on fully reusable vehicles with wings, and trade studies were performed on the benefits of horizontal vs vertical takeoff. These studies, however, were performed independently separate NASA contractors.

A.0.2 The Mid 1960's

The fresh flow of ideas certainly served as a basis for the mid-60's design tasks. In 1965, General Dynamics evaluated a lifting body second stage with two different first stage configurations: airbreathing and all rocket first stage [45]. The airbreathing configuration evaluated the impact of Liquid Air Cycle Engine, as well as vertical or horizontal takeoff on the vehicle performance. Economic studies determined that rocket engines would be less risky and have a lower development costs than air-breathing engines. Nevertheless, the Air Force was still pushing for the TSTO with an airbreathing booster. As a result, three vehicle architectures made it through 1965: 1) Dynasoar, a lifting body flying atop a Saturn I-B, 2) a TSTO all rocket fully reusable booster and orbiter, and 3) a TSTO airbreathing booster and lifting body second stage - see Figure A.2. Even after five years of conceptual design studies, it was still unsure which architecture would make the final design.

Meanwhile, studies where ongoing at NASA from which the stage-and-a-half configuration emerged. This concept would use a reusable orbiter, flying with expendable tanks around it. Imagined by Max Hunter, a NASA engineer, this concept would comply with the development cost constraints without reaching the high cost per flight of fully expendable boosters. It had a higher cost per flight when compared to fully reusable concepts but a lower development cost. This new idea of the stage-and-a-half concept was going to be the origin of a new set of concepts that later emerged in 1967.

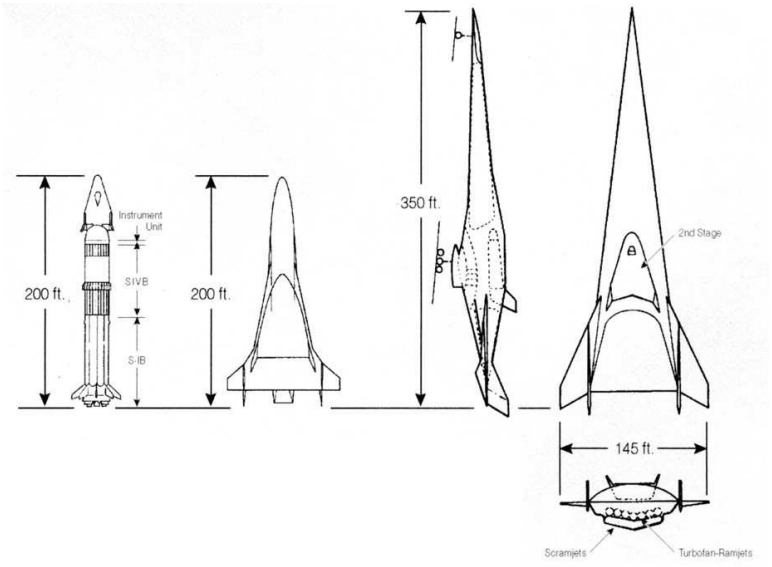
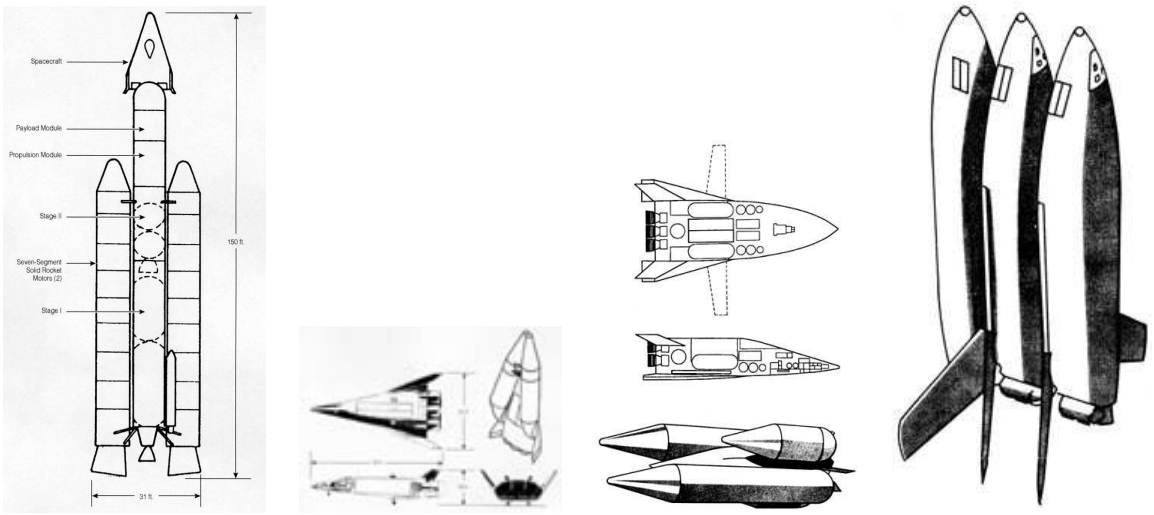


Figure A.2: Dynasoar, TSTO concept, and airbreathing concept. Source: Heppenheimer, 2002 [45]

At the end of this year, the concepts developed by a new round of funding to NASA contractors were presented to a symposium. Martin Marietta thought of a derivative of the Dynasoar presented in Figure FigDynasoar above. It was a lifting body flying atop a Titan III-M expendable launch vehicle - see Figure A.3. Conservative, this concept was designed to minimize the development cost and offer a launch vehicle system that lied on the expendable missile and launch vehicle heritage. Lockheed, though, believed in a more innovative configuration: Max Hunter's configuration of the Star Clipper, a single stage to orbit (SSTO) vehicle with a lifting body vehicle with expendable tanks fixed on each side of the body leading edges. Although not new, the studies surrounding the performance and economics evaluation of this concept served in bringing enough insights about SSTO vehicles. Similarly, McDonnell Douglas imagined the Tip-Tank concept. As shown in Figure A.3c, this concept had two propellant tanks attached on each side of the vehicle (between the wing and the tail) and one attached on top. The wings were deployable for landing. This concept



(a) Martin Marietta Titan-IIIM conctept (b) Lockheed Star-Clipper concept (c) McDonnell Douglas Tip Tank concept (d) General Dynamics Triamese concept

Figure A.3: Most popular concepts by the end of 1966. Source: Heppenheimer, 2002 [45]

however, like the Star-Clipper would be not be totally reusable because of the expendable tanks fixed on the vehicle side. Unlike the three other contractors, General Dynamics developed the Triamese - see Figure A.3d. This fully reusable concept consisted of three identical stages fixed together. The middle vehicle would deliver the payload to orbit whereas the two other vehicles were the liquid rocket boosters. This innovative configuration would reduce the manufacturing cost due to a higher number of identical units produced when compared to TSTO or SSTO concepts.

While the Triamese approach was perceived as innovative, another endeavor brought recognition to General Dynamics. The company had in fact performed an architecture comparison of the existing configurations. From the Titan III-M to the TSTO fully reusable vehicle, General Dynamics compared five different architectures based on their development cost and cost per flight [45]. This study confirmed the belief that fully reusable concepts, although cheap in recurring cost, have much higher development cost when compared to expendable or half-expendable concepts. In addition, the study demonstrated the importance of selecting a launch vehicle concept by comparing its possible

configurations with the same set of performance tools.

A.0.3 The Late 1960's and Early 1970's

With NASA's goal to decrease the cost per launch to that of an airline type, the study presented by General Dynamics in Section A.0.2 made NASA redirect again its focus toward fully reusable concepts. Consequently, until the first half of 1969, each contractor was sent back to the drawing boards in order to perfect one concept. McDonnell Douglas focused on the Tip-Tank and fully reusable concepts, Lockheed on the Star-Clipper and the Triamese, General Dynamics on the Triamese, and North American Rockwell on expendable boosters. Among NASA and its contractors, however, no organized design process existed. Each contractor was performing analyses on one or two launch vehicle concepts with their own performance analysis model, and each contractor and NASA engineer had their own preference for the most promising concept.

Nevertheless, knowledge about hypersonic reentry technologies was advancing. Debates on important issues were still ongoing: delta or straight wings, aluminum or titanium structures, hot structure or thermal protection tiles. Inside NASA, a new concept emerged: Faget's concept named after its creator Max Faget, a NASA aerodynamicist. His concept, illustrated in Figure A.4, consisted of a TSTO fully reusable vehicle where both vehicles, with straight wings for the return flight, would perform a no-lift reentry. To Faget, this concept was less risky, simple to calculate, and posed no difficulty to size since the aerodynamics of straight wings vehicles was well understood.

Many people inside NASA disliked this concept because of its simplicity, and of its shift in the center of lift for the different flight regimes. The opponents to Faget's concept favored delta wings instead. Although the approach speed was higher than straight wing vehicles, and the aerodynamics was not as well understood, the delta

Figure A.4: Max Faget's concept. Source: Heppenheimer, 2002 [45]

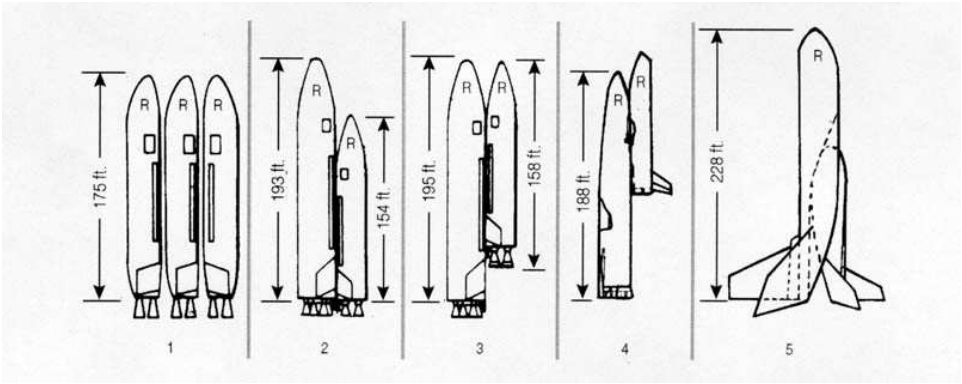
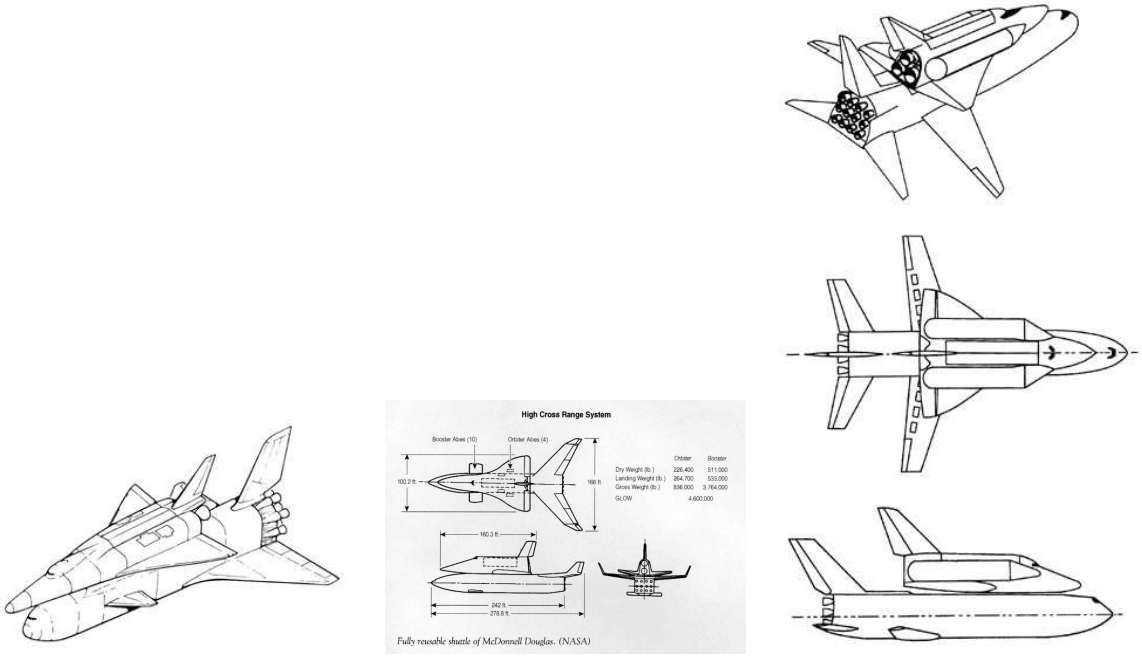


Figure A.5: Reusable concepts of 1969. Source: Heppenheimer, 2002 [45]

wing would offer better transition in the center of lift, generate less drag, and offer space for the landing gear. It would also produce more lift at hypersonic speeds and generate a greater amount of cross-range, which was an important Air Force requirement for reconnaissance missions [45]. The Delta wing would, however, be heavier, lead to a higher approach speed at landing and be more difficult to size. This debate nonetheless brought back the level of creativity that helped to generate more potential derivatives concepts. As a result, by the end of 1969, several new architectures were imagined and evaluated - see Figure A.5. This fixed the shuttle design to a TSTO fully reusable concept where the orbiter would be ignited after separation from the booster.

This new design contributed to a new wave of concepts generation and individual



(a) North American Rockwell fully reusable concept

(b) McDonnell fully reusable concept for cross range

(c) Grumman fully reusable concept

Figure A.6: Most popular concepts by the beginning of 1970. Source: Heppenheimer, 2002 [45]

studies performed by several NASA contractors. North American Rockwell first envisioned its fully reusable vehicle such as depicted in Fig. (A.6a). Its orbiter thermal protection system (TPS) was being meticulously manufactured with an aluminum and titanium structure mixture. McDonnell Douglas and Grumman came up with a different architecture shape illustrated in Figure A.6b and A.6c, respectively. McDonnell pushed for a configuration that would offer more cross range, whereas Grumman offered the expendable tank concept. For all these configurations, both the first and second stage would contain their own propellants.

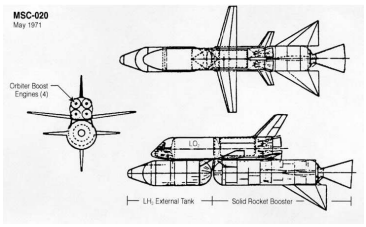
NASA experienced a major problem with the contractors concepts; it was in fact difficult to compare them because of the different assumptions and analysis methods used. An internal report published by Grumman stated:
"for those who have in the past undertaken to compare configurations from several contractors, [..] there is nothing more frustrating and inaccurate than to attempt to

compare weight, performance, and cost from several contractors, using, by definition, their own unique preliminary design ground rules and criteria". [45]

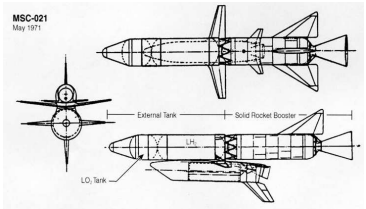
This observation stimulated the contractor to perform comparisons of concepts with the same set of models and assumptions. It resulted in a comparison study of 29 different shuttle configurations. The results of this analysis helped to draw three major conclusions. First, that a fully reusable shuttle was better financially in the long run. However, if budget constraints such as the peak funding were taken into account, other alternatives were more promising. Second, that the McDonnell, Grumman, and Rockwell's concepts would not obtain funding because of their high development cost. Third, that expendable tankage would reduce the development cost and the cost per flight [45].

Stimulated by these results, Max Faget derived a new series of space shuttles with external tanks based on the fully reusable and stage-and-a-half concepts. By then, the orbiter shape was determined as being a delta wing with blunt nose body. Nevertheless the orbiter and booster configurations were still under design. In May 1971 Max Faget first started by putting the booster liquid hydrogen in an expendable external tank atop the reusable booster. This concept, named MSC-020, would bring down the orbiter weight and staging velocity. He placed both the oxidizer (LOX) and fuel (LH2) in external tanks, always atop the booster stage (MSC-021, MSC-023) to further decrease the orbiter weight and staging velocity. Finally, he eliminated the booster stage and put all the propellant in an expendable external tank fixed under the orbiter (MSC-040). This new concept led to a further significant decrease in the orbiter weight and staging velocity [45]. This last concept was selected as the baseline vehicle in September 1971.

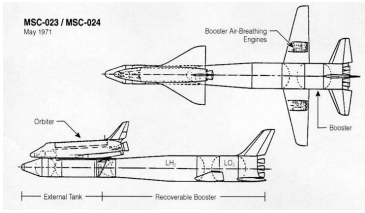
This concept MSC-040 pressured the last phase of conceptual design, which consisted in selecting the booster stage architecture. Until then, the booster had always been seen as a reusable stage composed mainly of the propellant tanks and rocket



(a) MSC-020 concept of expendable LH2 tank



(b) MSC-021 concept of expendable LH2 and LOX tanks atop an expendable solid rocket booster stage



(c) MSC-023 concept of expendable LH2 and LOX tanks atop a reusable solid rocket booster stage

Figure A.7: Evolution of the expendable tank concepts. Source: Heppenheimer, 2002 [45]

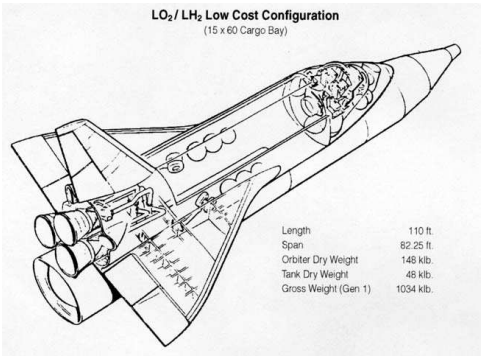


Figure A.8: MSC-040 concept of expendable external tank with reusable orbiter. Source: Heppenheimer, 2002 [45]

engines. Not developed yet, this type of booster would just add on the already constrained development cost. In mid-1971, however, Martin Marietta had come up with the concept of a six solid rocket boosters strapped on a reusable first stage propelled by liquid rocket engines. This concept was to reduce the development cost and minimize the risk of the entire program by using developed technology (i.e., the solid boosters). At the same time, NASA Marshall came up with a pressure-fed expendable booster. This was to eliminate the utilization of turbomachinery and therefore reduce the peak-funding required to develop the turbopumps. But, their use would signify a drop in the performance and an increase in the booster and propellant weights. Nevertheless, a more realistic cost analyses concluded that these pressure-fed boosters were still violating the budget constraints in development cost fixed by NASA.

It was only few months after that thrust-assisted shuttle concepts arose. It consisted of a thrusting method where two stages operate in parallel. Many configurations of this concept existed but the most promising one was an orbiter strapped-on to an external LOX and LH2 tank that would fly in parallel with two solid or pressure-fed liquid recoverable boosters. This concept seemed to please most of the NASA officials because it represented a reduction in development cost. Pressure-fed boosters were preferred over solid booster because they offered better performance. It was then selected as the new baseline by the end of 1971.

The last step consisted of presenting the shuttle concept to President Nixon to get his approval. Although NASA had been concentrated on the thrust assisted stage-and-a-half concept, it presented two alternatives to congress. The first was a TSTO fully reusable vehicle. The second was the stage-and-a-half concept with four booster options: 1) fully reusable with wing LOX/LH2 with serial burning , 2) pressure-fed with LOX/LH2 first stage burning in serial, 3)thrust assisted with two pressure-fed boosters, or 4)thrust assisted with two solid boosters. There was a strong preference for pressure-fed liquid rocket boosters. This preference came from

NASA Marshall and its leaders, the V-2 rocket designers. They knew that liquid engines had a better performance than the solid rocket, they would allow an easier flight control management during the vehicle ascent, and they would rely on Saturn's space program technology. However, as the orbiter design details were increasing, so became an increase in cost estimation. The latest analyses showed that the orbiter was going to cost more than expected which left no choice for the booster selection but to choose the cheapest, that is, the solid rocket boosters. The pressure-fed or pump-fed boosters would go over the $5.5B budget constraint, which would not have been accepted by congress. The solid boosters, however, given a much smaller development cost, would allow the program to stay within the budget limits. Consequently, after more than a decade of conceptual design, the final configuration was selected to be a delta wing orbiter, flying on an expendable LOX-LH2 expendable tank with two solid rocket boosters burning in parallel with the shuttle liquid engines. This architecture was not to be changed and led to today's Space Shuttle architecture as depicted in Figure A.9.

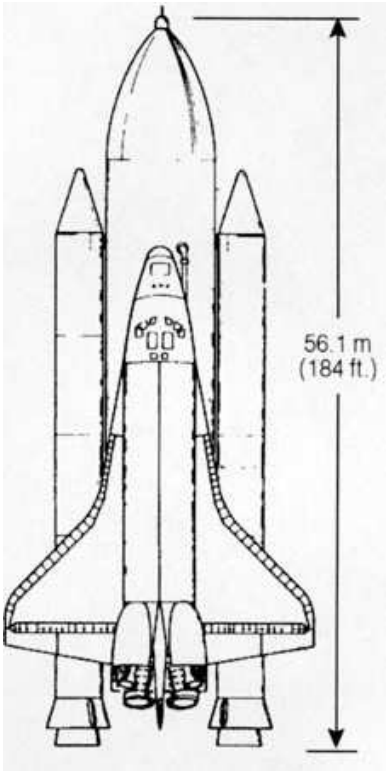


Figure A.9: Final conceptual design for the space shuttle. Source: Heppenheimer, 2002 [45]

APPENDIX B

TRAJECTORY RSES

- DV1 = 29122.7136284939190 + -4913.3384519608708 * 1/(1+exp(-1*(-8.8938849702005 + 2.2973679922741 * T2W_liftoff + 0.0175569586437 * Isp_stage_1))) + -3259.3854206811693 * 1/(1+exp(-1*(1.1444745413131 + 2.8349726948758 * T2W_liftoff + -0.0138597616086 * Isp_stage_1))) + 4792.1821619760540 * 1/(1+exp(-1*(16.6592008020938 + -14.6178742880638 * T2W_liftoff + 0.0050654456909 * Isp_stage_1))) + 4710.4671586131326 * 1/(1+exp(-1*(-16.8860671397752 + 9.7666337717400 * T2W_liftoff + 0.0123867420817 * Isp_stage_1))) + 1888.6714595591834 * 1/(1+exp(-1*(-5.9728237792917 + 4.8948298100574 * T2W_liftoff + -0.0020068452816 * Isp_stage_1)))

Figure B.1: Neural Network equation for the ΔV required of single stage launch vehicles without boosters

Table B.1: Response surface coefficients for the booster stage ΔV required of single stage launch vehicles with boosters

Parameter	**Coefficient**
Intercept	1.390E+03
$V_{staging-boosters}$	7.735E-01
$Isp_0 \cdot Isp_0$	6.837E-03
$V_{staging-boosters} \cdot Isp_0$	-5.499E-04
$V_{staging-boosters} \cdot Isp_1$	2.167E-04

Table B.2: Response surface coefficients for the core stage ΔV required of single stage launch vehicles with boosters

Parameter	Coefficient
Intercept	-6.668E+04
$MR_{boosters}$	-5.928E+03
Isp_0	8.011E+02
Isp_1	1.181E+02
$V_{staging-boosters}$	-1.231E+01
$MR_{boosters} \cdot MR_{boosters}$	3.974E+02
$Isp_1 \cdot Isp_0$	-4.083E-01
$V_{staging-boosters} \cdot Isp_0$	-8.340E-02
$V_{staging-boosters} \cdot V_{staging-boosters}$	2.589E-03

Table B.3: Response surface coefficients for the first stage ΔV required of two stage launch vehicles without boosters

Parameter	Coefficient
Intercept	1.002E+04
$(T/W)_{liftoff}$	-9.249E+03
$(T/W)_2$	-7.032E+02
Isp_1	7.760E+00
$V_{staging-stage1}$	9.508E-01
$(T/W)_{liftoff} \cdot (T/W)_{liftoff}$	2.828E+03
$Isp_1 \cdot (T/W)_{liftoff}$	-2.859E+00
$V_{staging-stage1} \cdot (T/W)_{liftoff}$	3.865E-02
$V_{staging-stage1} \cdot (T/W)_2$	4.705E-02
$V_{staging-stage1} \cdot Isp_1$	-1.374E-04

Table B.4: Response surface coefficients for the second stage ΔV required of two stage launch vehicles without boosters

Parameter	Coefficient
Intercept	3.014E+04
$(T/W)_2$	-5.090E+03
Isp_2	1.430E+01
$V_{staging-stage1}$	-1.694E+00
$Isp_2 \cdot (T/W)_2$	-9.343E+00
$V_{staging-stage1} \cdot (T/W)_2$	4.881E-01
$V_{staging-stage1} \cdot Isp_2$	-4.637E-04
$V_{staging-stage1} \cdot V_{staging-stage1}$	1.440E-05

Table B.5: Response surface coefficients for the booster stage ΔV required of two stage launch vehicles with boosters

Parameter	**Coefficient**
Intercept	4.963E+03
$(T/W)_{liftoff}$	-9.303E+03
$(T/Wfraction)_{boosters}$	9.342E+03
$(T/W)_2$	-4.202E+03
Isp_1	8.539E+00
Isp_2	2.257E+00
$V_{staging-boosters}$	3.211E-01
$V_{staging-stage1}$	-6.160E-02
$(T/W)_{liftoff} \cdot (T/W)_{liftoff}$	5.430E+03
$(T/Wfraction)_{boosters} \cdot (T/W)_{liftoff}$	-8.562E+03
$(T/Wfraction)_{boosters} \cdot (T/Wfraction)_{boosters}$	2.250E+03
$(T/W)_2 \cdot (T/Wfraction)_{boosters}$	1.536E+03
$Isp_0 \cdot MR_{boosters}$	1.268E-01
$Isp_0 \cdot (T/W)_{liftoff}$	-3.793E+00
$Isp_0 \cdot (T/Wfraction)_{boosters}$	1.627E+00
$Isp_0 \cdot (T/W)_2$	3.513E+00
$Isp_1 \cdot MR_{boosters}$	-1.300E-01
$Isp_1 \cdot (T/W)_{liftoff}$	-2.526E+00
$Isp_1 \cdot (T/W)_2$	4.450E+00
$Isp_1 \cdot Isp_1$	-8.947E-03
$Isp_2 \cdot Isp_2$	-3.490E-03
$V_{staging-boosters} \cdot (T/W)_{liftoff}$	-3.309E-01
$V_{staging-boosters} \cdot (T/Wfraction)_{boosters}$	7.475E-01
$V_{staging-boosters} \cdot Isp_1$	3.159E-04
$V_{staging-boosters} \cdot V_{staging-boosters}$	-1.583E-05
$V_{staging-stage1} \cdot Isp_0$	1.263E-04

```
DV1 = 8755.4269382360708 + 32021.1143641291890 * 1/(1+exp(-1*( 0.0616421793711 + -
   0.0387465644755 * mass_ratio_boosters +  2.8988401801234 * T2W_liftoff + -
   3.2517628803728 * T2W_boosters + -0.3025249771830 * T2W_stage_2 + -0.0036402288367
   * Isp_boosters +  0.0011419249637 * Isp_stage_1 +  0.0009828713815 * Isp_stage_2 +
   0.0001113627667 * Staging_velocity_boosters + -0.0000523100388 *
   Staging_velocity_stage_1))) + 49278.6793031753770 * 1/(1+exp(-1*( 1.9002285327308 +
   -0.0962941394288 * mass_ratio_boosters + -3.7886774520518 * T2W_liftoff +
   3.7783463902430 * T2W_boosters + -1.1421545705912 * T2W_stage_2 +  0.0005284251339
   * Isp_boosters + -0.0009876066389 * Isp_stage_1 + -0.0005403339268 * Isp_stage_2 +
   -0.0003208910855 * Staging_velocity_boosters +  0.0000735280158 *
   Staging_velocity_stage_1))) + 22052.0880637503070 * 1/(1+exp(-1*(-7.0273538264960 +
   -0.1429086260173 * mass_ratio_boosters +  4.4241910492697 * T2W_liftoff + -
   1.5225410469741 * T2W_boosters +  2.8293540171548 * T2W_stage_2 + -0.0115022250780
   * Isp_boosters + -0.0046693456036 * Isp_stage_1 +  0.0008085020775 * Isp_stage_2 +
   0.0004452067513 * Staging_velocity_boosters +  0.0000790698982 *
   Staging_velocity_stage_1))) + -29775.5435212098930 * 1/(1+exp(-1*( 1.9799455900557
   + -0.0413679249879 * mass_ratio_boosters +  3.0886706821242 * T2W_liftoff + -
   2.7084719417781 * T2W_boosters + -0.3734292411094 * T2W_stage_2 + -0.0040711697923
   * Isp_boosters + -0.0011721056158 * Isp_stage_1 +  0.0008702137571 * Isp_stage_2 +
   0.0001649349757 * Staging_velocity_boosters + -0.0002005659962 *
   Staging_velocity_stage_1))) + 47654.9227418317750 * 1/(1+exp(-1*(-2.5743790102760 +
   0.0343600836141 * mass_ratio_boosters +  6.9088459527059 * T2W_liftoff + -
   5.3036523637246 * T2W_boosters + -0.3718562555905 * T2W_stage_2 + -0.0021540116452
   * Isp_boosters + -0.0142140793350 * Isp_stage_1 +  0.0000555342737 * Isp_stage_2 +
   0.0002489047314 * Staging_velocity_boosters + -0.0002229768554 *
   Staging_velocity_stage_1)))
```

Figure B.2: Neural Network equation for the first stage ΔV required of two stage launch vehicles with boosters

```
DV2 = 37964.1390767279880 + -32589.1058221799540 * 1/(1+exp(-1*(-0.5821765929413 +
    0.0894462663057 * mass_ratio_boosters +  7.3334541908164 * T2W_liftoff + -
    6.0993196654872 * T2W_boosters + -1.5694551169956 * T2W_stage_2 + -0.0021124926108
    * Isp_boosters + -0.0150923296935 * Isp_stage_1 +  0.0027922284947 * Isp_stage_2 +
    0.0003556107738 * Staging_velocity_boosters + -0.0003830725513 *
    Staging_velocity_stage_1))) + -12093.6006894516690 * 1/(1+exp(-1*(-2.6285329688161
    + -0.2032078065102 * mass_ratio_boosters + -5.2025482322857 * T2W_liftoff +
    5.2225211504416 * T2W_boosters +  5.2524118266421 * T2W_stage_2 + -0.0019759878829
    * Isp_boosters +  0.0166399200329 * Isp_stage_1 + -0.0067837084949 * Isp_stage_2 +
    -0.0003575243846 * Staging_velocity_boosters +  0.0004512436998 *
    Staging_velocity_stage_1))) + -31275.6419065607260 * 1/(1+exp(-1*(-2.4275892809478
    +  0.0026321230758 * mass_ratio_boosters + -0.0658062484444 * T2W_liftoff +
    0.0734432867854 * T2W_boosters +  0.1761453593305 * T2W_stage_2 +  0.0001342989220
    * Isp_boosters +  0.0000346269651 * Isp_stage_1 + -0.0000344964075 * Isp_stage_2 +
    -0.0000023588808 * Staging_velocity_boosters +  0.0001534087025 *
    Staging_velocity_stage_1)))
```

Figure B.3: Neural Network equation for the second stage ΔV required of two stage launch vehicles with boosters

APPENDIX C

LAUNCH VEHICLE WEIGHT BREAKDOWN

Table C.1: CaLV booster mass breakdown results from validation

Mass group	**Mass, lbm**
1.0 Body	189,549
2.0 Primary power	2,023
3.0 Electrical conversion and distribution	13,809
4.0 Hydraulics	2,427
5.0 Margin	15,793
Dry mass (1 to 10)	223,601
Propellant	1,416,220
Payload	908,052
Gross liftoff mass	1,639,822
Ascent mass ratio	1.78

Table C.2: CaLV first stage mass breakdown results from validation

Mass group	**Mass, lbm**
1.0 Body	108,391
1.1 Fuel tank	36,724
1.2 Oxidizer tank	23,162
1.3 Insulation	23,162
1.4 Antivortex and slash baffles	2,967
1.5 Thrust structure	17,289
1.6 Intertank	13,329
1.7 Interstage	4,143
1.8 Forward and aft skirts	2,320
1.9 Engine compartment	3,277
2.0 Main propulsion	57,130
2.1 Engines	33,164
2.2 Engines installation	1,393
2.2 Engines subsystem	1,393
2.3 TVC	2,947
2.4 Feeding system	11,102
2.5 Purge system	6,160
2.6 Pressurization system	970
3.0 TPS/TMS	6,496
4.0 OMS /RCS	0
5.0 Primary power	4,549
6.0 Electrical conversion and distribution	0
7.0 Hydraulics	1,738
8.0 Actuators	0
9.0 Avionics	670
10.0 Margin	14,318
11.0 Residual propellants	16,712
12.0 OMS/RCS reserve propellants	0
13.0 Ascent reserve propellants	3,332
14.0 Inflight losses and vents	263
15.0 Ascent propellants	2,221,607
16.0 Startup losses	0
Dry mass (1 to 10)	193,293
Propellant	2,241,915
Payload	654,444
Gross liftoff mass	3,177,062
Ascent mass ratio	3.60

Table C.3: CaLV second stage mass breakdown results from validation

Mass group	Mass, lbm
1.0 Body	21,901
1.1 Fuel tank	9,446
1.2 Oxidizer tank	3,840
1.3 Insulation	3,840
1.4 Antivortex and slash baffles	621
1.5 Thrust structure	1,125
1.6 Intertank	4,405
1.7 Interstage	0
1.8 Forward and aft skirts	1,347
1.9 Engine compartment	0
2.0 Main propulsion	12,583
2.1 Engines	7,404
2.2 Engines installation	298
2.2 Engines subsystem	298
2.3 TVC	631
2.4 Feeding system	2,643
2.5 Purge system	1,078
2.6 Pressurization system	231
3.0 TPS/TMS	2,098
4.0 OMS/RCS	0
5.0 Primary power	1,749
6.0 Electrical conversion and distribution	0
7.0 Hydraulics	416
8.0 Actuators	0
9.0 Avionics	430
10.0 Margin	3,510
11.0 Residual propellants	5,319
12.0 OMS/RCS reserve propellants	5
13.0 Ascent reserve propellants	629
14.0 Inflight losses and vents	59
15.0 Ascent propellants	459,233
16.0 Startup losses	0
Dry mass (1 to 10)	42,686
Propellant	465,241
Payload	133,703
Gross mass (including payload)	654,431
Ascent mass ratio	3.50

Table C.4: Best TSTO architecture first stage mass breakdown resulting from an ACO simulation

Mass group	**Mass, lbm**
1.0 Body	152,381
1.1 Fuel tank	14,334
1.2 Oxidizer tank	35,790
1.3 Insulation	35,790
1.4 Antivortex and slash baffles	6,363
1.5 Thrust structure	67,959
1.6 Intertank	13,324
1.7 Interstage	2,070
1.8 Forward and aft skirts	1,866
1.9 Engine compartment	6,550
2.0 Main propulsion	137,066
2.1 Engines	68,609
2.2 Engines installation	4,011
2.2 Engines subsystem	4,011
2.3 TVC	8,488
2.4 Feeding system	44,580
2.5 Purge system	3,409
2.6 Pressurization system	3,958
3.0 TPS/TMS	12,589
4.0 OMS /RCS	14,707
5.0 Primary power	7,215
6.0 Electrical conversion and distribution	19,905
7.0 Hydraulics	6,257
8.0 Actuators	0
9.0 Avionics	503
10.0 Margin	28,050
11.0 Residual propellants	28,471
12.0 OMS/RCS reserve propellants	0
13.0 Ascent reserve propellants	6,541
14.0 Inflight losses and vents	516
15.0 Ascent propellants	4,360,479
16.0 Startup losses	0
Dry mass (1 to 10)	378,671
Propellant	4,396,006
Payload	1,103,342
Gross liftoff mass	6,040,784
Ascent mass ratio	3.91

Table C.5: Best TSTO architecture second stage mass breakdown resulting from the ACO simulation

Mass group	**Mass, lbm**
1.0 Body	41,723
1.1 Fuel tank	16,045
1.2 Oxidizer tank	10,033
1.3 Insulation	10,033
1.4 Antivortex and slash baffles	1,157
1.5 Thrust structure	2,348
1.6 Intertank	5,209
1.7 Interstage	0
1.8 Forward and aft skirts	2,503
1.9 Engine compartment	2,983
2.0 Main propulsion	19,344
2.1 Engines	11,306
2.2 Engines installation	475
2.2 Engines subsystem	475
2.3 TVC	1,005
2.4 Feeding system	4,122
2.5 Purge system	1,601
2.6 Pressurization system	360
3.0 TPS/TMS	3,960
4.0 OMS/RCS	2,598
5.0 Primary power	1,952
6.0 Electrical conversion and distribution	4,461
7.0 Hydraulics	828
8.0 Actuators	0
9.0 Avionics	323
10.0 Margin	6,737
11.0 Residual propellants	8,704
12.0 OMS/RCS reserve propellants	0
13.0 Ascent reserve propellants	1,174
14.0 Inflight losses and vents	110
15.0 Ascent propellants	856,576
16.0 Startup losses	0
Dry mass (1 to 10)	81,925
Propellant	866,564
Payload	133,703
Gross mass (including payload)	1,103,291
Ascent mass ratio	4.76

APPENDIX D

TECHNOLOGY SELECTION PROBLEM

D.1 Technology Selection

The technology selection is a problem examined for the application of the new design method. It is a problem that is widely known in the field of aerospace engineering [84, 100]. Therefore, it is an excellent verification problem because the new design process can be compared to a more traditional approach. Moreover, this problem shows, for the first time, the use of ACO as an optimization algorithm for the technology selection problem. ACO usually solves combinatorial problems such as the Traveling Salesman Problem (TSP), the knapsack problem, and the job scheduling problems with great efficiency [28]. It is thus interesting to see how the algorithm performs in a different context.

The goal of the problem is to find, under budget constraints, the best technology portfolio from a set of available technologies. The design method presented in Figure 3.2 is applied and compared to a traditional approach of selecting a technology portfolio, as presented by Roth, et al. [83] Three different optimization algorithms are compared: Ant Colony Optimization (ACO), Genetic Algorithm (GA), and Simulated Annealing (SA). The comparison is done to verify the effectiveness of ACO in selecting technologies, compared to traditional GA or SA optimization approaches.

The next sections describe how the new design process is implemented on the technology selection problem. The results obtained from the new design process are then compared with the results obtained from SA and GA.

Table D.1: Responses used in the technology selection problem

Number	Response	Baseline Value	Units
R1	Lift-to-drag ratio (Mach .85, 40kft)	19.21	-
R2	Weight engine	234,843	lbs
R3	Total sideline noise	94.9	db
R4	Total takeoff noise	92.0	db
R5	Total approach noise	98.3	db
R6	TSFC (Mach .85, 35kft)	0.5588	lbs/lbf-hr
R7	Engine thrust-to-weight ratio	3.92	-
R8	NOx emission	53.8	lbs
R9	Weight fuel	85,294	lbs
R10	TOGW	659,025	lbs
R11	Direct operating cost	0.04348	$/pass.-mile
R12	Landinf field length	5828	ft
R13	Takeoff Field length	9534	ft
R14	Approach speed	122.7	fps
R15	Acquisition cost	119.68	$M

D.1.1 Step 1: Define the Design Objectives and Mission Requirements

The first step of the new concept exploration process is to define the requirements of the problem. For this problem the design objectives consist of minimizing an Overall Evaluation Criterion (OEC), described in Equation D.1,

$$OEC = R_1/R_{ref_1} + \sum_{i=2}^{n} (R_i/R_{ref_i}) \tag{D.1}$$

where R_i represents a response of the system (or objective), and R_{ref_i} the baseline value of a response i. It can be observed that R_1 is computed differently from the other responses because it must be maximized, whereas the other responses are minimized. The problem considers 15 responses, measuring the aircraft performance, weight, noise, emissions, and costs. Each response varies as a function of the technology combinations. The responses are described in the Table D.1, with their respective baseline value.

Table D.2: Technologies considered for the technology selection problem

#	Technology	#	Technology
T1	High Speed Slotted Wing	T16	HQ Tube/Liner Integration
T2	Transonic Adaptive Bump	T17	Low NOx Combustor Development on GE Engines
T3	Sensory Materials and Damage Science	T18	Low NOx Combustor Development on P&W Engines
T4	ST Manufacturable Large Structures	T19	3000F CMC combustor materials
T5	Slat-Cover Filler	T20	$3000°F$ metallic combustor materials
T6	Landing-Gear Noise Reduction	T21	2 Stage Proof of Concept Compressor
T7	Core Cowl Acoustic Liner	T22	High Pressure Turbine
T8	Installation Improved Chevron Nozzles	T23	LPT with aggressive duct
T9	Flap Trailing Edge Treatment for Jet Interaction	T24	Fan Containment
T10	Soft Vanes (Stators)	T25	Ni Disk - Gayda
T11	Fan Duct Acoustic Splitter	T26	Lightweight Single Crystal Blade Alloy
T12	Offset Stream Technologies	T27	Low Conductivity TBC
T13	Chevron Vortex Stabilization	T28	$2700°F$ CMC Liner
T14	Fluidic Chevrons	T29	$2700°F$ CMC Vane
T15	Inlet Blowing/Liner Integration		

D.1.2 Step 2: Select a set of concepts and technologies

The second step of the design method consists of selecting a pool of concepts and technologies that could potentially fulfill the design objectives mentioned in the previous section. As mentioned earlier, this problem regards technologies only, which explains why no concept is included in the process. Thus, a pool of 29 promising technologies is selected, in hope of improving the overall aircraft performance. Each technology is described in Table D.2. The technologies range from aerodynamic improvement technologies (e.g., T1, T2, T13, T14, and T15), material and structure technologies (e.g.,T3, T4, and T9) to engine technologies (e.g., T17 to T29). Some technologies, however, are incompatible which is modeled by a technology incompatibility matrix, presented in the next section.

D.1.3 Step 3: Populate the Morphological and Incompatibility Matrix

The third step consists of organizing the concepts and technologies in the morphological matrix. The problem does not require a morphological matrix because it only considers technologies.

The problem has three different types of technology incompatibility, as shown in Table D.3. The pairwise incompatibilities (first column) are modeled by the incompatibility matrix, whereas the other incompatibilities are addressed via a penalty function.

Table D.3: Technology incompatibilities

Technology Incompatibility Type		
Select 1 out of 2	**Select 2 out of 3**	**Select 3 out of 4**
T11 and T16	T8, T12 and T13	T8, T12, T13, and T14
T15 and T16	T8, T12 and T14	
T15 and T24	T8, T13 and T14	
T16 and T24	T13, T12 and T14	
T17 and T18		
T19 and T20		
T19 and T28		
T20 and T28		

D.1.4 Step 4: Map the Technologies to the Model's Design Variables

The fourth step of the method consists of mapping the concepts and technologies to the input variables of the modeling environment. Thus, the impact of the 29 technologies is assessed by the technology impact matrix, using the approach shown in Equation D.2.

$$[K]^{60\mathbf{X}1} = [TIM]^{60\mathbf{X}29} \cdot [S_T]^{29\mathbf{X}1} \tag{D.2}$$

For Equation D.2, $[K]$ is the k-factors vector, $[TIM]$ is the technology impact matrix, and $[S_T]$ is the technology string. The technology string is the design vector where each bit is associated to a technology and it has a value of 1 if the technology is "on" or 0 otherwise. A total of 60 k-factors are chosen to model the performance of the technology. Each technology models a change into a number of k-factor values, to form the technology impact matrix, presented in Tables D.5 and D.5. For example, Technology 1 (High speed slotted wing), impacts the wing sweep by increasing the

critical Mach number, the aerodynamic tech factor, and the takeoff drag coefficient. Thus, the technology enables the attainment 26.05° for the sweep angle, 2.18 for the aerodynamic tech factor, and a 4.5% reduction in the takeoff drag.

Table D.4: Technology impact matrix for the 29 technologies

	k1	k2	k3	k4	k5	k6	k7	k8	k9	k10	k11	k12	k13	k14	k15	k16	k17	k18	k19	k20	k21	k22	k23	k24	k25	k26	k27	k28	k29	k30
T1																													26.81	2.15
T2																														
T3																														
T4																														
T5																														
T6																														
T7																														
T8																												1.00		
T9																														
T10																														
T11																												0.99		
T12																												1.00		
T13									0.00																					
T14									0.02																					
T15																														
T16																														
T17		1.48		1.30		25.00				0.72	0.85																			
T18		1.48		1.58		25.00				0.70	0.85																			
T19											0.98	-0.69																		
T20											0.98																			
T21					-0.01		1.32	2.02																						
T22														0.00	0.20															
T23																					0.01	0.02	0.28							
T24																														
T25																														
T26																78.00	78.00	78.00	-0.02	-0.02				94.00	94.00	-0.03	-0.03			
T27																200	200	200						200	200					
T28											0.94	-0.69																		
T29													3650			700														

Table D.5: Technology impact matrix for the 29 technologies (continued)

	k31	k32	k33	k34	k35	k36	k37	k38	k39	k40	k41	k42	k43	k44	k45	k46	k47	k48	k49	k50	k51	k52	k53	k54	k55	k56	k57	k58	k59	k60
T1		-0.04	-0.02																											
T2	0.98																													
T3					-0.12	-0.42						0.07																		
T4				-0.24	-0.18							-0.17	0.03																	
T5			-0.02				0.07					0.05																-1.00		
T6								0.01				0.00																	-3.00	
T7																								-0.25						
T8																									-2.00	-1.50	-1.40			
T9																														-1.00
T10																					-1.00	-3.00								
T11																					0.00	-1.00								
T12																											-2.00			
T13																											-0.50			
T14																											-1.00			
T15																			-2.00											
T16																			-2.50	-2.50				-2.50						
T17																														
T18																														
T19																														
T20																														
T21																														
T22																														
T23																														
T24																-1.00	-1.00	-1.00												
T25																														
T26																														
T27																														
T28																														
T29																														

D.1.5 Step 5: Generate Compatible Architectures

The fifth step of the design process consists of generating compatible architectures. In the case of the technology selection problem, it consists of generating compatible technology sets. The generation of compatible technology sets is simpler to perform than a traditional architecture selection problem. This is because the design space exploration only involves technologies, which do not have as many incompatibilities as

the concepts usually do. The next paragraph explains how technologies are selected.

As mentioned earlier, the goal of this technology selection problem is to compare the traditional approach with the new design method. Thus far, the first four steps are similar but Step 5 is the point where the traditional and new design processes differentiate themselves. On the one hand, the traditional approach optimizes the design space using a GA and SA. The population members are initially generated using a uniformly random vector of 0 and 1. The population member are then changed with the algorithm through genetic operators (GA) and random swaps (SA). The compatibility of the objective function is checked by using a penalty function, which worsens the objective if the solution is incompatible. On the other hand, the new design process construct compatible solutions using a graph, which 29 nodes represent the 29 technologies. This approach is used by ACO, and the incompatibilities are modeled by constraining the solution to certain edges on the graph. This approach is repeated until the architecture is completed. The next step consists of evaluating the architecture performance, which is discussed in the next section.

D.1.6 Step 6: Sizing the Architecture

The sixth step of the design process consists of sizing the architecture generated in Step 5. This operation requires the computation of Equation D.2, which computes the performance from the input k-factors. Once the k-factors are computed, the aircraft performance is evaluated through RSEs, as shown in Equation D.3,

$$R = RSE\left([K]\right) \tag{D.3}$$

where R is the response, RSE is the response surface equation, which is function of the k-factor vector $[K]$. Equation D.3 represents the procedure required to evaluate the performance of each technology set. In this case, however, the model is a meta-model and is represented by RSEs. Those RSEs compute the value of the 15 responses

(R), which are used in the overall evaluation criterion (Step 1) to determine the overall vehicle performance. This value is then used by the design space exploration algorithm, as detailed in the next section.

D.1.7 Design Space Exploration

The design space exploration is an internal process by which solutions are created and examined repeatedly, tying Steps 5 and 6 together. As mentioned in the introductory paragraph, the design space exploration is performed with three stochastic optimization approaches: 1) Ant Colony Optimization, 2) Genetic Algorithm, and 3) Simulated Annealing. These three optimization approaches are selected because they enable the optimization over a discrete design space. The convergence rate and optimality of each algorithm is compared to determine which technique is better suited for this technology selection problem.

For comparison purposes, each technique is assigned similar optimization parameters. Thus, each algorithm is set for maximum number of 30 iterations with a population of 20 members. Moreover, the ACO algorithm has an evaporation rate of 0.01, the GA has crossover and mutation fractions of 0.8 and 0.01, respectively, and the simulated annealing has an initial temperature of 800° and a cooling rate of 0.8. A summary of the optimization settings is presented in Table D.6 for the ACO, Table D.7 for the GA, and Table D.8 for the SA.

Table D.6: Ant Colony Optimization settings for technology selection validation problem

Optimization parameter	Setting
Number of ants	20
Number of iterations	30
Pheromone evaporation rate	0.01

Table D.7: Genetic Algorithm settings for technology selection problem

Optimization parameter	Setting
Population size	20
Number of generations	30
Number of best individuals sent to next generation	2
Crossover fraction	0.8
Mutation fraction	0.01

Table D.8: Simulated Annealing settings for technology selection problem

Optimization parameter	Setting
Initial temperature	800
Cooling rate	0.8
Maximum number of iteration	20
Maximum number of repetition	30

D.1.8 Analysis of the Results

The results from the design space exploration are shown in this section. First, Figure D.1 shows the variation of the objective function value as a function of the iteration number for the three stochastic optimization algorithms. The different starting point for each algorithm arise from the initial randomly selected population for each algorithm at iteration 0. Iteration represents the best overall objective function value once each algorithm has performed its changes in the initial population. This figure also shows that the ACO finds a slightly better solution than GA, and a much better solution than SA. Moreover, the same figure shows that ACO converges faster than SA or GA. Overall, this simulation shows that SA performs poorly for this problem compared to GA and ACO. These results of course depend on the initial settings of the three algorithms but the settings used here are typical to combinatorial optimization problems.

Since stochastic optimization algorithms have a certain level of randomness in their algorithm, the outcome from the optimization can vary from one simulation to the other. For this reason, the optimization shown in Figure D.1 is repeated 250

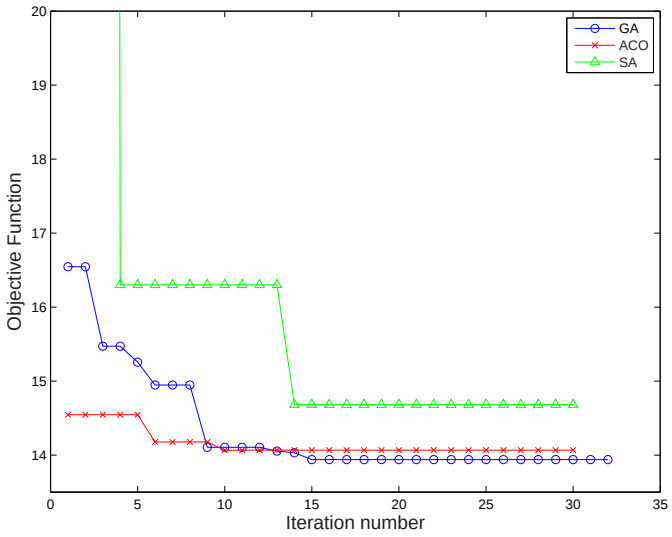
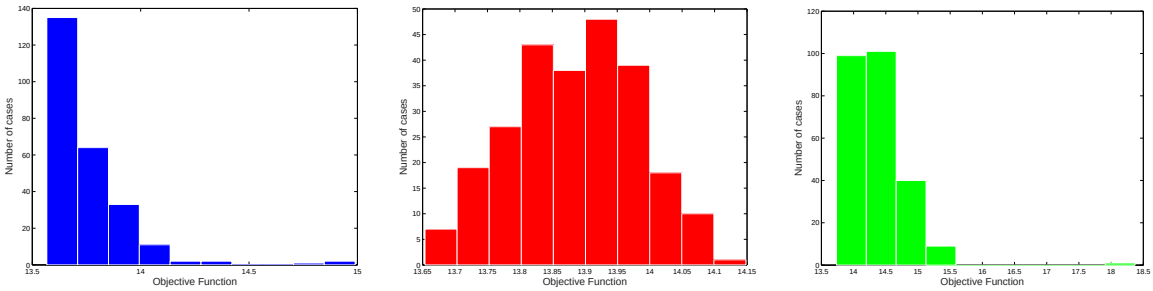


Figure D.1: Variation of the objective function value as a function of the iteration number

times to compare the algorithms based on their convergence rate and relative error. Figure D.2 shows the distribution of the value of the objective function over the 250 cases and Table D.9 shows the mean and standard deviation for each algorithm. This shows that ACO has a better mean and standard deviation than GA, and that SA does again poorly for the optimization of this problem. However, GA found the best overall value of 13.56 during the 250 repetitions, which means that it explores more widely the design space.

Moreover, one can observe the different distribution shapes from each optimization algorithm. Indeed, SA and GA have a distribution thats looks logarithmic, whereas ACO's distribution looks more normal. Figure D.1.8 compares the three algorithms with their respective calculated frequency distribution to show the same observations as above. This figure is the same as Figure D.2 except that bars are replaced by dots to compare the distribution of each algorithm on a same graph.

Another important criterion for optimization of combinatorial design spaces is the convergence rate. To assess this metrics, the number of iterations required for convergence is analyzed in Figure D.4. This figure shows the distribution of convergence rate



(a) GA (b) ACO (c) SA

Figure D.2: Distribution of the objective function for the 250 cases

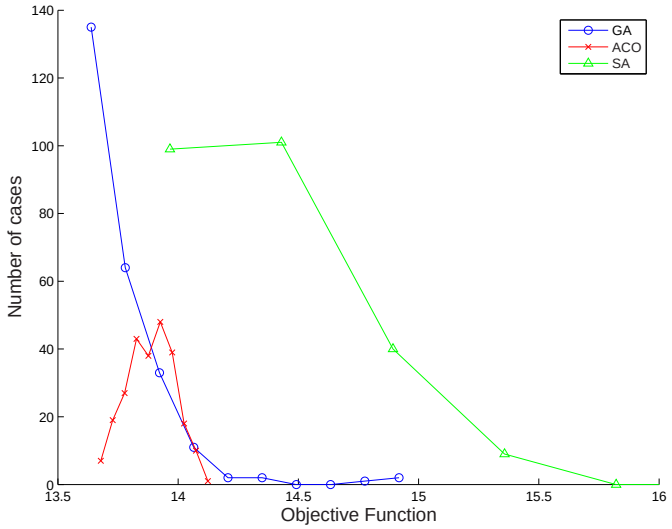
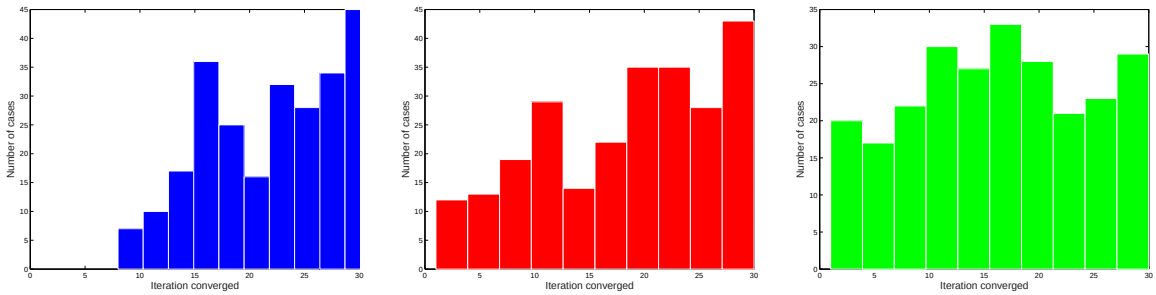


Figure D.3: Summary of the distribution of the objective function for the 250 cases

Table D.9: Comparison between the performance of ACO, GA, and SA for the technology selection problem

	ACO	GA	SA
Mean	14.16	14.36	33.53
Standard deviation	0.178	0.751	55.70



(a) GA (b) ACO (c) SA

Figure D.4: Distribution of the objective function value for the GA, ACO, and SA

Table D.10: Summary of the results for the technology selection problem

	Ranking	
Algorithm	Convergence rate	Relative error
ACO	# 1	# 2
GA	# 2	# 1
SA	# 3	# 3

for all three optimization algorithms. The graphs thus show that ACO and SA have a better convergence rate over GA. However, because SA does so poorly in finding good solutions, the convergence rate for this algorithm becomes meaningless.

D.1.9 Findings

First, it is observed that ACO converges faster than GA and SA. This observation comes from the graph of Figure D.1, which shows that ACO attains quicker a near-optimal solution than the other two algorithms. Second, in terms of relative error ACO averages better final solutions than GA and SA but GA tends to get better overall solutions. This is observed from the data presented in Figure D.4 and Table D.9. Therefore, it seems that GA explores more widely the design space, hence its large standard deviation, and therefore the algorithm is more likely to find the optimum value of the solution. In light of these comments, Table D.10 presents a ranking of the three methods analyzed over the convergence rate and the convergence error.

REFERENCES

[1] "Exploration systems architecture study," tech. rep., 2005.

[2] Baker, A. P. and Mavris, D. N., "Assessing the simultaneous impact of requirements, vehicle characteristics, and technologies during aircraft design," AIAA Paper 2001-0533, 2001.

[3] Baker, A. P., *The Role of Mission Requirements, Vehicle Attributes, Technologies and Uncertainty in Rotorcraft System Design*. PhD thesis, Georgia Institute of Technology, 2003.

[4] Beckers, R., Deneubourd, J.-L., and Goss, S., "Modulation of trail laying in the ant lasius niger (hymenoptera: Formiciday) and its role in the collective section of a food source.," *Journal of Insect Behavior*, vol. 6, no. 6, pp. 751–759, 1993.

[5] Blasi, L., Iuspa, L., and Core, G. D., "Conceptual aircraft design based on a multiconstraint genetic optimizer," *Journal of Aircraft*, vol. 37, no. 2, pp. 351–354, 1999.

[6] Bowcutt, K. G., "Multidisciplinary optimization of airbreathing hypersonic vehicles," *Journal of Propulsion and Power*, vol. 17, pp. 1184–90, Nov.-Dec. 2001.

[7] Bowcutt, K. G., Gonda, M., Hollowell, S., and Ralston, T., "Performance, operational and economic drivers of reusable launch vehicles," AIAA Paper 2002-3901, 2002.

[8] Braun, R. D., Powell, R. W., Lepsch, R. A., Stanley, D. O., and Kroo, I. M., "Comparison of two multidisciplinary optimization strategies for launch vehicle design," *Journal of Spacecraft and Rockets*, vol. 32, no. 3, pp. 404–10, 1995.

[9] Brothers, B., "Launch vehicle mass estimating relationship database," tech. rep., Marshall Space Flight Center, 1999.

[10] Buckley, M. J., Fertig, K. W., and Smith, D. E., "Design sheet: An environment for facilitating flexible trade studies during conceptual design," AIAA Paper 92-1191, 1992.

[11] Buonanno, M. A., *Aircraft Concept Exploration using Multicriteria Interactive Genetic Algorithm*. PhD thesis, Georgia Institute of Technology, 2005.

[12] Buonanno, M. A. and Mavris, D. N., "Aerospace vehicle concept selection using parallel,variable fidelity genetic algorithms," AIAA Paper 2004-4432, 2004.

[13] Caluori, V., Conrad, R., and Jenkins, J., "Technology requirements for future earth-to-geosynchronous orbit transportation systems," Tech. Rep. CR-3265, NASA, 1996.

[14] Chase, R. L., "A comparative assessment of reusable launch-vehicle candidates," AIAA Paper 79-894, 1979.

[15] Cheng, F. Y. and Li, D., "Genetic algorithm development for multiobjective optimization of structures," *AIAA Journal*, vol. 36, pp. 1105–1112, June 1998.

[16] Chung, H.-S., "Supersonic business jet design using a knowledge-based genetic algorithm with an adaptive, unstructured grid methodology," AIAA Paper 2003-3791, 2003.

[17] Colorni, A., Dorigo, M., Maffioli, F., Maniezzo, V., Righini, G., and Trubian, M., "Heuristics from nature for hard combinatorial optimization problems," *International Transaction in Operational Research*, vol. 3, no. 1, pp. 1–21, 1996.

[18] Cook, W., "www.tsp.gatech.edu (traveling salesman problem)," September 2006.

[19] Crocker, A. M., Charania, A., and Olds, J. R., "Introduction ro rosetta models and the aties process for advanced space transportation technology investment." SpaceWorks Engineering White Paper, 2005.

[20] Crocker, A. M., Charnia, A., and Olds, J. R., "An introduction to the rosetta modeling process for advanced space transportation technology investment," No. 2001-4625, 2001.

[21] Crossley, W., Martin, E. T., and Fanjoy, D. W., "A multiobjective investigation of 50-seat commuter aircraft using genetic algorithm," AIAA Paper 2001-5247 2001.

[22] Dasgupta, D. and McGregor, D. R., "sga: A structured genetic algorithm," tech. rep., Department of Computer Science, University of Strathclyde, 1993.

[23] DeBaets, P. and Mavris, D. N., "Aeroelastic design by combining conventional practice with bi-level integrated system synthesis," AIAA Paper 2004-4431, 2004.

[24] Dell'Amico, M., Maffioli, F., and Martello, S., *Annotated Bibliographies in Combinatorial Optimization*. John Wiley & Sons, 1997.

[25] Divan, P., *Aerodynamic Preliminary Analysis System II - Part 2.* NASA, 1981. NAS1-15674.

[26] Doerner, K., Gutjahr, W. J., Hartl, R. F., Strauss, C., and Stummer, C., "Ant colony optimization in multiobjective portfolio selection," July 2001.

[27] Dorigo, M., Maniezzo, V., and Colorni, A., "The ant system: Optimization be a colony of cooperation agent," *IEEE Transactions on Systems*, vol. 26, no. 1, pp. 1–13, 1996.

[28] Dorigo, M. and Stutzle, T., *Ant Colony Optimization.* MIT Press, 2004.

[29] Dorigo, M., *Optimization, learning and natural algorithms (in Italian).* PhD thesis, Dipartimento di Elettronica, 1992.

[30] Dorrington, G., "Comparisons of rocket and air-breathing vehicle concepts for earth-to-orbit transportation," AIAA Paper 90-2702, 1990.

[31] Ehrgott, M. and Gandibleux, X., "A survey and annotated bibliography of multi-objective combinatorial optimization," *OR Spektrum*, vol. 22, pp. 425–460, 2000.

[32] Finger, S. and Dixon, J. R., "A review of research in mechanical engineering design. part 1.," *Research in Engineering Design*, vol. 1, pp. 121–137, 1989.

[33] Fogler, H. S., *Strategies for Creative Problem Solving.* Prentice-Hall PTR, 1995.

[34] Freeman, D. C., Talay, T. A., Stanley, D. O., and Wilhite, A., "Design options for advanced manned launch systems," AIAA Paper 90-3816, 1990.

[35] Glatt, C. and Hague, D., "Odin: Optimal design integration system," Tech. Rep. CR–2462, NASA, 1975.

[36] Glover, F., "Tabu search: A tutorial," *Interfaces*, vol. 20, no. 4, pp. 74–94, 1990.

[37] Goss, S., Aron, S., Deneubourg, J.-L., and Pasteels, J. M., "Self-organized shrotucts in the argentine ant," *Naturwissenschafen*, vol. 76, pp. 570–581, 1989.

[38] Group, S. E. H. W., *Systems Engineering Handbook.* International Council on Systems Engineering (INCOSE), 2a ed., June 2004.

[39] Gutjahr, W. J., "S-aco: An ant-based approach to combinatorial optimization under uncertainty," pp. 238–249, Sept. 5-8 2004.

[40] Haefeli, R., Littler, E., Hurley, J., and Winter, M. G., "Technology requirements for advanced earth-orbital transportation systems," Tech. Rep. CR-2866, NASA, 1977.

[41] Hartsfield, N. and Ringel, G., *Pearls in Graph Theory: a comprehensive introduction*. Academic Press Ltd, 1994.

[42] Haskins, C., *Systems Engineering Handbook*. International Council on Systems Engineering, June 2006.

[43] Hassan, R. A. and Crossley, W. A., "Multi-objective optimization of communication satellites with two-branch tournament genetic algorithm," *Journal of Spacecraft and Rockets*, vol. 40, pp. 266–272, March - April 2003.

[44] Hepler, A. K. and Bangsund, E. L., "Technology requirements for advanced earth orbital transportation system. (volume 1 - executive summary)," tech. rep., NASA, 1978.

[45] Heppenheimer, T. A., *History of the Space Shuttle*, vol. 1 & 2. Smithsonian Institution Press, 2002.

[46] Hoffman, K. and Padberg, M., "Combinatorial and integer optimization," April 2007. http://iris.gmu.edu/ khoffman/.

[47] Hollingsworth, P. M., *Requirements Controlled Design: A Method for Discovery of Discontinuous System Boundaries in the Requirements Hyperspace*. PhD thesis, Georgia Institute of Technology, 2004.

[48] Hollingsworth, P. M. and Mavris, D. N., "A method for concept exploration of hypersonic vehicles in the presence of open and evolving requirements," 2000.

[49] Huang, X., "A prototype computerized synthesis methodology for generic space access vehicle conceptual design," AIAA Paper 2005-2539, 2005.

[50] Humble, R. W., Henry, G. N., and Larson, W. J., *Space Propulsion Analysis and Design*. McGraw-Hill, 1995.

[51] Inc., S. E., "Nasa-air force cost model (nafcom) ddt&e and tfu parametric cost estimating relationships." Spreadsheet.

[52] Jackson, S., *Systems Engineering for Commercial Aircraft*. Asgate, 1997.

[53] Jilla, C. D. and Miller, D. W., "Multi-objective, multidisciplinary design optimization methodology for distributed satellite system," *Journal of Spacecraft and Rocket*, vol. 41, no. 1, pp. 39–50, 2004.

[54] Johnson, C. and Schutte, J., *Basic Regression Analysis for Integrated Neural Networks (BRAINN) Documentation*. Georgia Institute of Technology.

[55] Joyner, C. R. and Sabatella, J. A., "Launch vehicle propulsion optimization using response surface methodology," AIAA Paper 1990-2433, 1990.

[56] Kirby, M. R., *A Methodology for Identification, Evaluation, and Selection in Conceptual and Preliminary Aircraft Design*. PhD thesis, Georgia Institute of Technology, 2001.

[57] Kirby, M. R. and Mavris, D. N., "Forecasting technology uncertainty in preliminary aircraft design," Oct. 19-21 1999.

[58] Kirby, M. R. and Mavris, D. N., "A method for technology selection based on benefit, available schedule and budget resources," Oct. 10-12 2000.

[59] Kirby, M. R., *Saving Time with Automated Response Surfaces (STARS)*. Georgia Institute of Technology, 2006.

[60] Koelle, D. E., "Handbook if cost engineering for space transportation systems," tech. rep., TransCostSystems, 2003.

[61] Lawler, E. and Wood, D., "Branch-and-bound methods: A survey," *Operations research*, vol. 14, no. 4, pp. 699–719, 1966.

[62] Mattson, C. A. and Messac, A., "Concept selection in n-dimension using s-pareto frontiers and visualization," AIAA Paper 2002-5418, 2002.

[63] Mattson, C. A. and Messac, A., "Development of a pareto-based concept selection method," AIAA Paper 2002-1231, 2002.

[64] Mavris, D. N., "Interactive reconfigurable morphological matrix." Internal Document, Aerospace Systems Design Lab, 2006.

[65] McManus, H. L., Hastings, D. E., and Warmkessel, J. M., "New methods of rapid architecture selection and conceptual design," *Journal of Spacecraft and Rockets*, vol. 41, pp. 20–28, 2004.

[66] Morton, D., "Fundamentals of modern systems engineering." Presentation, 2006.

[67] Mosher, T., "Conceptual spacecraft design using a genetic algorithm trade selection process," *Journal of Aircraft*, vol. 36, pp. 200–208, Jan - Feb 1999.

[68] Mullur, A., Mattson, C. A., and Messac, A., "New decision matrix based approach for concept selection using linear physical programming," AIAA Paper 2003-1446, 2003.

[69] NASA, *Hypersonic Air Vehicle Optimization Code Users Guide*. NASA Ames Research Center, 2001.

[70] NASA Report CR-114986, *Space Shuttle Synthesis Program - Volume 1*, 1970.

[71] Nickol, C. and Guynn, M., "High altitude long endurance air vehicle analysis of alternatives and technology requirements development," AIAA Paper 2007-1050, 2007.

[72] Olds, J. R., Crocker, A., Bradford, J., and Charania, A., "Acre-92 ssto eto rlv." Presentation, April 2001. Concept Documentation.

[73] Patel, M., Lewis, K. E., Maria, A., and Messac, A., "System design through subsystem selection using physical programming," *AIAA Journal*, vol. 41, pp. 1089–1096, June 2003.

[74] Perez, R. E., Chung, J., and Behdinan, K., "Aircraft conceptual design using genetic algorithms," AIAA Paper 2000-4938, 2000.

[75] Powell, R., Striepe, S., Desai, P., and Braun, R., *Program to Optimize Simulated Trajectories (POST)*. NASA, NAS1-13611, October 1997.

[76] Pugh, S., *Creating Innovative Products Using Total Design: The Living Legacy of Stuart Pugh*. Addison-Wesley, 1st ed., 1996. 544 p.

[77] Raczynski, C. M., Lewe, J. H., Kirby, M. R., and Mavris, D. N., "Technology portfolio assessments using a gene-correction genetic algorithm," AIAA Paper 2003-6731, 2003.

[78] Reddy, S. Y., Fertig, K. W., and Smith, D. E., "Constraint management methodology for conceptual design tradeoff studies," August 18-22 1996.

[79] Reingold, E. M., Nievergelt, J., and Deo, N., *Combinatorial Algorithms: Theory and Practice*. Prentice-Hall, 1977.

[80] Ritchey, T., "Modeling complex socio-technical systems using morphological analysis." Adapted from an address to the Swedish Parliamentary IT Commission, 2002.

[81] Ritchey, T., "Strategic decision support using computerized morphological analysis," 14-16 Sept. 2004.

[82] Rohrschneider, R. R., "Development of mass estimating relationship database for launch vehicle conceptual design." Georgia Institute of Technology, 2002.

[83] Roth, B., German, B., and Mavris, D., "Adaptive selection of engine technology solution sets from a large combinatorial space," AIAA Paper 2001-3208 2001.

[84] Roth, B. and Patel, C., "Application of genetic algorithms in the engine technology selection process," 2003.

[85] Rowell, L. F., Braun, R. D., Olds, J. R., and Unal, R., "Multidisciplinary conceptual design optimization of space transportation systems," *Journal of Aircraft*, vol. 36, pp. 218–26, Jan.-Feb. 1999.

[86] Shonkwiler, R., "Monte carlo methods class notes." Class notes, 2004.

[87] Smaling, R. M. and de Weck, O. L., "Fuzzy pareto frontiers in multidisciplinary system architecture analysis," AIAA Paper 2004-4553, Aug. 2003.

[88] Sobieszczanski-Sobieski, J., "Multidisciplinary aerospace design optimization survey of recent development," AIAA Paper 96-0711, 1996.

[89] Sobieszczanski-Sobieski, J., Agte, J. S., and Sandusky, R. R., "Bi-level integrated system synthesis (bliss)," pp. 1543–1557, AIAA Paper 98-4916 1998.

[90] Sobieszczanski-Sobieski, J., Emiley, M. S., Agte, J. S., and Sandusky, R. R., "Advancement of bi-level integrated system synthesis (bliss)," AIAA Paper 2000-0421, 2000.

[91] Sobieszczanski-Sobieski, J. and Kodiyalam, S., "Bliss/s - a new method for two-level structural optimization," *40th Structures, Structural Dynamics, and Materials Conference and Exhebit*, vol. 2, pp. 1274–1286, 1999.

[92] Stanley, D. O., *Technology Engineering: A Systems Engineering Approach to the Concurrent Development of Space Transportation Systems and Technology.* PhD thesis, George Washington University, 2002.

[93] Suh, N. P., *The Principles of Design.* Oxford University Press, 1990.

[94] Sun, Z.-G. and Teng, H.-F., "Optimal layout design of a satellite module," *Engineering Optimization*, vol. 35, no. 5, pp. 513–529, 2003.

[95] Sutton, G. P., *Rocket Propulsion Elements.* Wiley-Interscience, 6th ed., 1992.

[96] Szedula, J. A., "Fastpass: A tool for launch vehicle synthesis," AIAA Paper 96-4051, 1996.

[97] Thomson, W. T., *Introduction to Space Dynamics.* Wiley & Sons, 1986.

[98] Tucker, A., *Applied Combinatorics.* John Wiley and Sons, 3rd ed., 1995.

[99] Ulungu, E. L. and Teghem, J., "Multi-objective combinatorial optimization problems: A survey.," *Journal of Multi-Criteria Decision Analysis*, vol. 3, pp. 83–104, 1994.

[100] Utturwar, A., Rallabhandi, S., DeLaurentis, D., and Mavris, D., "A bi-level optimization approach for technology selection," AIAA Paper 2002-5426, Sept. 2002.

[101] Villeneuve, F. and Mavris, D. N., "A new method of architecture selection for launch vehicles," AIAA Paper 2005-3361, 2005.

[102] Wade, M., "Encyclopedia astronautica," August 2006.

[103] Weber, R. G. and Condoor, S. S., "Conceptual design using a synergistically compatible morphological matrix," *Frontiers in Education Conference*, 1998.

[104] Wilhite, A. W. and Rehder, J. J., "Avid: A design system for technology studies of advanced transportation concepts," AIAA Paper 79-0872, 1979.

[105] Wood, R. M. and Bauer, X. S., "A discussion of knowledge based design," AIAA Paper 98-4944, 1998.

[106] Yokohama, N. and Suzuki, S., "Modified genetic algorithm for constrained trajectory optimization," *ournal of Guidance, Control, and Dynamics*, vol. 28, pp. 139–144, Jan - Feb 2005.

[107] Zeke Strawbridge, Daniel A. McAdams, R. B. S., "A computational approach to conceptual design," 2002.

[108] Zwicky, F., "The morphological method of analysis and construction," 1948.

[109] Zwicky, F. and Wilson, A. G., *New Methods of Thought and Procedure; Contributions on Symposium on Methodologies*. Springer-Verlag, 1967.